

CIRCUIT CELLAR

INK®

# THE COMPUTER APPLICATIONS JOURNAL

February 1994 — Issue #43

## APPLICATIONS IN LIGHT

Photonic Transistors Grow  
Into Optical Computers

Image Display Using  
the LightStick II

Networkable Intelligent  
Thermostat

Assembly Language  
Program Design Technique



\$3.95 U.S.  
\$4.95 Canada

# EDITOR'S INK

## Warp Speed, Mr. LaForge



Imagine your desktop computer running at the speed of light. Sure, the electrons that are coursing their way through your present machine are already traveling close to that speed, but those pesky little

switches sitting on a bed of silicon really slow the works down. Rev those up to the speed of light as well, and you'll have a real screamer on your hands.

Fantasy? Not necessarily, and you'll have your chance to help make it a reality.

We first ran an article on the photonic transistor about a year-and-a-half ago that described this new little wonder and some of its potential. Now we go the next step in our first article this month and look at its potential for use in full-blown computers. Many problems remain to be worked out, so Intel doesn't have to worry about the Pentium quite yet, but when these little holograms become real, they could change the face of computing. Find out what work is being done and how you can get involved.

I'm particularly fond of our second article because I was there when the germ of the idea was planted. I first saw the original LightStick at the Exploritorium while I was visiting Shaun with some other college buddies about seven years ago. Since then, we've both built LightSticks (though mine pales in comparison) and I've had feedback from other Circuit Cellar readers about their own experiences with the concept after they saw the Wavy Sign, one of our Honorable Mentions in the Fifth Annual Circuit Cellar Design Contest (by the way, this article was already in the works when that contest entry came in, so it was a complete coincidence). Take a look and have some fun.

Do you automatically fill in the interrupt vector table when you sit down to develop a new embedded project? Are interrupts a way of life, no matter what you're trying to do? If so, you may want to rethink your philosophy. Depending on what you're trying to do, living without them can make your life much easier, as our third article tries to show.

Next, the great debate over whether to use assembly language or a high-level language continues, but what about a compromise? Our last feature article describes a programming technique that combines the best of both into a discipline that many would do well to adopt.

In our columns, Ed continues talking about how to add a large LCD panel to his Firmware Development system with a discussion of the hardware. Jeff adds some smarts to the new DS1620 thermostat chip to make a networkable, intelligent thermostat. Tom crosses enemy lines into the analog camp to look at some new chips that are bound to give the venerable op-amp a run for its money. John introduces a new series in which he'll be describing how to build a general-purpose embedded control system based on a compact backplane. Finally, Russ looks at patents that deal with our theme of Applications in Light.

CIRCUIT CELLAR **INK**®

## THE COMPUTER APPLICATIONS JOURNAL

FOUNDER/EDITORIAL DIRECTOR  
Steve Ciarcia

EDITOR-IN-CHIEF  
Ken Davidson

TECHNICAL EDITOR  
Michael Swartzendruber

ASSOCIATE EDITOR  
Rob Rojas

ENGINEERING STAFF  
Jeff Bachiochi & Ed Nisley

WEST COAST EDITOR  
Tom Cantrell

CONTRIBUTING EDITORS  
John Dymbowski & Russ Reiss

NEW PRODUCTS EDITOR  
Harv Weiner

ART DIRECTOR  
Lisa Ferry

GRAPHIC ARTIST  
Joseph Quinlan

CONTRIBUTORS:  
Jon Elson  
Tim McDonough  
Frank Kuechmann  
Pellervo Kaskinen

Cover Illustration by Bob Schuchman  
PRINTED IN THE UNITED STATES

PUBLISHER  
Daniel Rodrigues

PUBLISHER'S ASSISTANT  
Susan McGill

CIRCULATION COORDINATOR  
Rose Mansella

CIRCULATION ASSISTANT  
Barbara Maleski

CIRCULATION CONSULTANT  
Gregory Spitzfaden

BUSINESS MANAGER  
Jeannette Walters

ADVERTISING COORDINATOR  
Dan Gorsky

CIRCUIT CELLAR INK, THE COMPUTER APPLICATIONS JOURNAL (ISSN 0896-8985) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Suite 20, Vernon, CT 06066 (203) 875-2751. Second class postage paid at Vernon, CT and additional offices. One-year (12 issues) subscription rate U.S.A. and possessions \$21.95, Canada/Mexico \$31.95, all other countries \$49.95. All subscription orders payable in U.S. funds only, via international postal money order or check drawn on U.S. bank. Direct subscription orders and subscription related questions to The Computer Applications Journal Subscriptions, P.O. Box 7694, Riverton, NJ 06077 or call (609) 786-0409. POSTMASTER: Please send address changes to The Computer Applications Journal, Circulation Dept., P.O. Box 7634, Riverton, NJ 06077

### HAJAR ASSOCIATES NATIONAL ADVERTISING REPRESENTATIVES

NORTHEAST  
**Debra Andersen**  
(617) 769-8950  
Fax: (617) 769-8982

SOUTHEAST  
**Christa Collins**  
(305) 966-3939  
Fax: (305) 985-8457

WEST COAST  
**Barbara Jones & Shelley Rainey**  
(714) 540-3554  
Fax: (714) 540-7103

MID-ATLANTIC  
**Barbara Best**  
(908) 741-7744  
Fax: (908) 741-6823

MIDWEST  
**Nanette Traetow**  
(708) 789-3080  
Fax: (708) 789-3082

Circuit Cellar BBS—24 Hrs. 300/1200/2400/9600/14.4k bps, 8 bits, no parity, 1 stop bit, (203) 871-1988; 2400/9600 bps Courier HST. (203) 871-0549

All programs and schematics in Circuit Cellar *INK* have been carefully reviewed to ensure their performance is in accordance with the specifications described, and programs are posted on the Circuit Cellar BBS for electronic transfer by subscribers

Circuit Cellar *INK* makes no warranties and assumes no responsibility or liability of any kind for errors in these programs or schematics or for the consequences of any such errors. Furthermore, because of possible variation in the quality and condition of materials and workmanship of reader-assembled projects, Circuit Cellar *INK* disclaims any responsibility for the safe and proper function of reader-assembled projects based upon or from plans, descriptions, or information published in Circuit Cellar *INK*.

Entire contents copyright © 1994 by Circuit Cellar Incorporated. All rights reserved. Reproduction of this publication in whole or in part without written consent from Circuit Cellar Inc is prohibited.

14

**Programmable Lightware/How to Develop Photonic Products for Fun and Profit**

*by John Hait*

26

**In the Eye of the Beholder/Displaying Messages with the LightStick II**

*by Shaun Greaney*

36

**Interrupt-free Design**

*by Do- While Tones*

40

**An Assembly Language Programming Aid**

*by Hank Wallace*

48

**Firmware Furnace**

Hardware for the '386SX Project's Bitmapped LCD Panel

*Ed Nisley*

58

**From the Bench**

Warming Up the Digital Thermostat

*Jeff Bachiochi*

64

**Silicon Update**

Op-amp Terminators

*Tom Cantrell*

70

**Embedded Techniques**

Data Logging and Collection Devices

*John Dybowski*

# INSIDE ISSUE 43

2

**Editor's INK**

Ken Davidson

**Warp Speed,  
Mr. LaForge**

6

**Reader's INK**

Letters to the Editor

8

**New Product News**

edited by Harv Weiner

77

**Patent Talk**

Russ Reiss

**ConnectTime**

Excerpts from  
the Circuit Cellar BBS  
conducted by  
Ken Davidson

**Steve's Own INK**

Steve Ciarcia  
And the Survey Says...

**Advertiser's Index**

83

96

81

# READER'S INK

## Not Quite So Basic Temperature Control

The "Home Temperature Control Basics" article in the January '94 issue was a fine analytical presentation. I enjoyed the analysis as well as the insight provided. I would like to ask that the author consider the significant energy savings that can be obtained by using electronic adjustable-speed drives (ASDs) to control the speed of motors that operate fans, blowers, and centrifugal pumps; that is, variable-torque loads.

The power to be saved over other flow control methods such as louvers, dampers, flow restrictors, bypass valves, and so forth is described by the cube of the reduced flow rate. For example, a 20% flow reduction using dampers for a fan would make the motor work harder and in no way reduce required horsepower, whereas speed/flow reduction using an ASD produces the following:  $PT(0.8)(0.8)(0.8) = 0.512PT + 0.05 = 0.562PT$ . We added the 5% to account for electrical losses in the drive. This principle is the basis for the electric utility rebate programs that are designed to encourage energy conservation and annually save millions of kilowatt hours.

For references, refer to "Adjustable Speed Drives Directory," third edition, from the Electric Power Research Institute, and "Electrical Engineering Pocket Handbook," 1993, from Electrical Apparatus Service Association. Thank you again for the fine article.

Joe Foster  
Fairport, N.Y.  
(via the Circuit Cellar BBS)

## Virtual Car Computer Request

I have been a devoted subscriber to your journal since its inception. Keep up the good work. While I might not have time to go through everything, I certainly use your product as a good reference from time to time. Have you thought of publishing a special index issue that gives a quick reference by subject, author, and other categorical information?

Anyway, I have just two questions I would like to pose to you, your staff, or perhaps some readers. First, where can I get information about the diagnostic codes and electrical interfaces on Ford computers? I bought a '93 Probe GT and I can't find anyone with this information. I know there are third-party vendors who sell computer interface devices and software for personal computers to perform diagnostic tests on cars equipped

with certain computer systems, but I can't find anyone who sells one for my vehicle. I think this is a very good area for future articles in your magazine.

Second, I am trying to get started in building my own virtual reality system at home. I have LCD shutter glasses for stereo viewing. I can't seem to find anyone who sells them for less than \$150 each. Is there anyone out there who manufactures just the glass alone? In other words, would it be cheaper for me to build my own instead of paying someone else to do something I am quite confident in doing myself. The trouble is finding the original source (again at a reasonable cost). Edmund Scientific sells this sort of material but the smallest size they sell is 6" x 6"-far too big for glasses.

Thanks.

Bert Schneider  
Colorado Springs, CO  
schneiderhg%dfanlab@pcmail.usafa.af.mil

*While we'll never publish a "special index issue," we do get a lot of requests for an article index. An index is an on-going project that we are working on as time permits, so there is no way of knowing when we'll have something completed (there is a lot of information in five years' worth of issues). In the meantime, one of the users of the Circuit Cellar BBS has put together a list that could be used as a makeshift index. The file is posted in the file area.*

*Your first question comes up on the BBS several times a year, but we have yet to hear any concrete answer to it. The auto manufacturers appear to keep a tight grip on the technical details on their on-board computers, giving it only to diagnostic tool manufacturers under nondisclosure. Plenty of people have been quick to provide the meanings of the "blink" codes used by some cars, but the amount of information that can be gleaned from those is woefully inadequate. If any of our readers can provide more information, we'll pass it along to everyone.*

*As for your second question, we can't provide any good answers here. Again, perhaps one of our readers can shed some light.*

## Giving Pitch Correction a KISS

Regarding the "Pitch Correction" discussion in the January '94 installment of "ConnecTime," Many years

# READER'S INK

ago I read about an invention developed at Bell Labs, I believe, to prevent audio feedback in public-address systems.

The principle was to shift the audio pitch by 5 Hz (cycles in those days) so the auditorium peak feedback frequency could not be sustained. The pitch change of 5 cycles (oops, Hertz) would be imperceptible by most ears.

The technique was to beat the audio up to an RF frequency by a local oscillator (to 455 kHz, I think) and then demodulate it with another oscillator running 5 Hz lower or higher.

A similar principle can be observed while listening to SSB (single sideband) AM radio. Tuning the receiver off of the center frequency changes the pitch drastically up or down. I seems to me that there is a simpler solution here that is less hardware/software intensive than using a DSP or CCD.

Warren L. Graves  
Georgetown, Calif.

## Contacting Circuit Cellar

We at the Computer *Applications Journal* encourage communication between our readers and our staff, so have made every effort to make contacting us easy. We prefer electronic communications, but feel free to use any of the following:

**Mail:** Letters to the Editor may be sent to: Editor, The Computer Applications Journal, 4 Park St., Vernon, CT 06066.

**Phone:** Direct all subscription inquiries to (609) 786-0409.

Contact our editorial offices at (203) 875-2199.

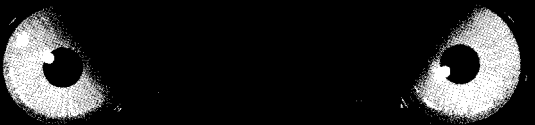
**Fax:** All faxes may be sent to (203) 872-2204.

**BBS:** All of our editors and regular authors frequent the Circuit Cellar BBS and are available to answer questions. Call (203) 871-1988 with your modem (300-14.4k bps, 8N1).

**Internet:** Electronic mail may also be sent to our editors and regular authors via the Internet. To determine a particular person's Internet address, use their name as it appears in the masthead or by-line, insert a period between their first and last names, and append "@circellar.com" to the end. For example, to send Internet Email to Jeff Bachiochi, address it to [jeff.bachiochi@circellar.com](mailto:jeff.bachiochi@circellar.com). For more information, send Email to [info@circellar.com](mailto:info@circellar.com).

"Is the

# TERROR



of **BIG S-L-O-W** software  
stalking your job security?"

Take some of the bytes out of your bloated code!

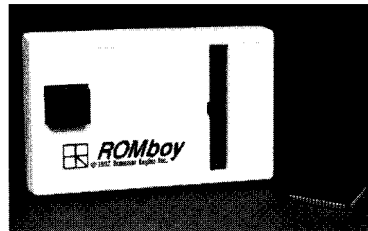
Call Franklin Software, the first and last  
word in quality for  
your 8051 and 80C166 software projects!

**F. FRANKLIN  
SOFTWARE, INC**

Call for your free Release VI Evaluation Kit!

tel: (408) 296-8051 fax: (408) 296-8061

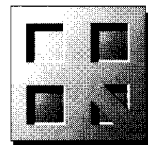
The BEST in ROM  
emulation technology:



- 1 Mbit
- 100ns
- ~~\$395~~ \$295

ROMboy includes a 10 day,  
no-risk money back guarantee!

Call Today — 800-776-6423



**Grammar  
Engine  
Inc.**

921 Eastwind Dr., Suite 122  
Westerville, OH 4308 1  
614/899-7878  
Fax 614/899-7888

Now Available  
4 Mbit

# ROMboy!

# NEW PRODUCT NEWS

Edited by Harv Weiner

## 87C750 MICROCONTROLLER DEVELOPMENT KIT

Developers working with the Philips 87C750 microcontroller can now get a low-cost, feature-packed tool set in a new software development kit available from Micro Computer Control Corporation. The **87C750-SDK** contains all the tools needed to create, assemble, and execute 87C750 programs.

**MICRO/EDITOR** is a full-featured, multiwindow text editor for creating and editing source programs. **MICRO/ASM-750** is a macro assembler that creates a robust programming environment that quickly finds and reports source file syntax errors, generates detailed program listings, and converts a source program into executable 87C750 format. **MICRO/SLD-750** is a simulator and source language debugger that can be used to execute 87C750 programs directly on a PC without any additional hardware.

This DOS-based development package has been specially configured to support an 87C750 target processor environment, including memory spaces, peripheral ports, and timers. Also supported is a source debug environment including instruction single-stepping, breakpoints, watch windows, and over 17 additional tools.



Applications for the 87C750 include consumer and industrial products that can benefit from its 80C51 architecture, small package size, and low cost. The 87C750-SDK is available until April 1994 for an introductory price of \$39.95. The regular list price is \$99.95.

Micro Computer Control Corp.  
P.O. Box 275.17 Model Ave. • Hopewell, NJ 08525  
(609) 466-1751 • Fax: (609) 466-4116

#500

## ENERGY SAVING DEVICE FOR PCS

A low-cost hardware device for reducing the energy consumption of PC-compatible and PS/2 personal computer peripherals has been announced by B&B Electronics. The Green Keeper connects to the keyboard of any PC. A monitor, printer, modem, or other peripheral can be plugged into Green Keeper's AC outlet. The software included with the system allows users to set the unit to turn off connected peripherals after a user-specified period of inactivity. A simple touch of the keyboard or mouse restores power to the monitor and peripherals. The CPU is not affected.

The main application of Green Keeper is to reduce the energy consumption of computing. According to the Environmental Protection Agency, 40 percent of the computer monitors in the U.S. are left on all the time; each of these monitors generates up to \$114 in electric bills per year. Green Keeper cuts these costs by turning off monitors or other peripherals in a way that's simple and convenient for the user.

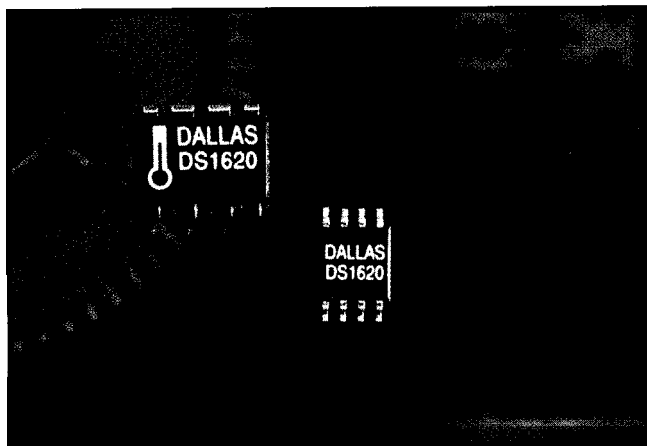
Green Keeper is contained in a 4.7" x 2" x 1.4" package and can handle a maximum load of 10 amps. Software is available for DOS, Windows 3.1, and OS/2 2.1. A Macintosh version is planned for release in early 1994. In keeping with environmental responsibility, the unit is packaged with 100% recycled and recyclable materials, and the software and documentation is on recycled disks. Green Keeper sells for \$69.95 and comes with a three-year warranty.



B&B Electronics Mfg. Co.  
4000 Baker Rd. • P.O. Box 1040, Ottawa, IL 61350 • (815) 434-0846 • Fax: (815) 434-7094

#501

# NEW PRODUCT NEWS



## DIRECT READING TEMPERATURE SENSOR

Dallas Semiconductor has announced an all-digital temperature sensor. Unlike other devices, which output an analog voltage that must be converted to a digital value, the DS1620 Digital Thermometer and Thermostat does direct digital temperature conversion. Temperature is converted to a 9-bit digital number in less than one second.

Fully calibrated at the factory, the DS1620 features 0.5° resolution over a -55°C to +125°C range and 0.5°C accuracy in the 0°C to 70°C range. Existing devices can

achieve this kind of accuracy only if they are calibrated in the end application.

The DS 1620 can also function as a stand-alone thermostat with two user-definable and changeable set points that are stored in nonvolatile memory. For example, the DS 1620 can eliminate excess power drain and noise in a portable computer by running the fan only when it is needed. Other applications for the DS1620 include environmental controls, industrial systems, consumer products, thermometers, and any thermally sensitive system.

Existing temperature sensors create a voltage drop across a nonlinear element and then have an external A/D converter measure the output. By contrast, the DS 1620 uses two oscillators whose frequency is determined by a resistor. The resistor in each oscillator has different temperature characteristics. The DS1620 determines temperature by comparing the values of the frequencies from the two oscillators.

The DS 1620 Digital Thermometer and Thermostat is available in an 8-pin DIP or 8-pin SOIC package and sells for \$2.50 in 5000-piece quantities.

Dallas Semiconductor  
4401 South **Beltwood** Pkwy. . Dallas, TX 75244  
(214) 450-0448 . Fax: (214) 450-0470

#502

## SOFTWARE IMPROVES MONITOR QUALITY

**DisplayMate** for Windows, a revolutionary software product that improves monitor picture quality, has been announced by Sonera Technologies. The program is said to improve sharpness and contrast, reduce certain forms of geometric distortion, minimize or eliminate annoying moire patterns, and improve color and grayscale accuracy.

DisplayMate for Windows works by presenting a slide show of screen images, similar in principle to the test patterns used by television stations. They have been scientifically designed to search for every possible potential weakness of a computer monitor at very high sensitivity. DisplayMate for Windows then shows how to improve the image, with the eye as the final judge of the results. This product requires no video knowledge and is appropriate for computer novices and experts alike.

Each screen contains a detailed description that explains "what to look for" and includes detailed step-by-step instructions on "what you can do" to improve the appearance of the image. In most cases, several different techniques are provided, each with an approach dependent on the nature of the problem and the user controls available on the monitor and video adapter. As an example of the new and innovative approach taken by the product: some types of geometric distortion can be reduced or eliminated by using the monitor's position control together with the video adapter's own image position adjustment in an offsetting combination. If the monitor and video adapter don't include these controls, alternative methods are suggested.

DisplayMate for Windows supports all color modes from monochrome up through full 24-bit color, including the popular 16-, 256-, and 16.8M-color modes. An extensive set of test patterns is provided for evaluating and exploring color quality, balance, range, and accuracy. For users interested in color matching, a special test pattern is provided for adjusting and matching both reflective and transmissive color samples using a number of color modes. DisplayMate for Windows sells for \$79.

Sonera Technologies . P.O. Box 565 . Rumson, NJ 07760 . (908) 747-6886 • Fax: (908) 747-4523

#503

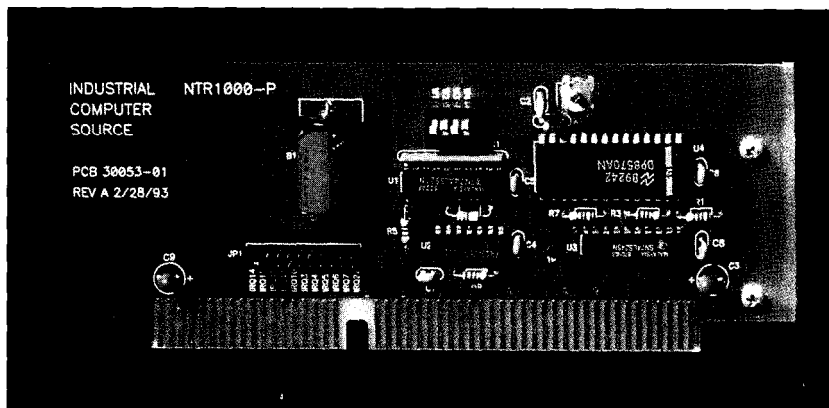
# NEW PRODUCT NEWS

## REAL-TIME CLOCK FEATURES HIGH ACCURACY

Industrial Computer Source has announced a low-cost real-time clock card for IBM PC and compatible computers. The **NTR1000-P** provides a stability of  $\pm 5$  seconds per month—better than two-fold improvement over standard DOS clocks—in harsh operating environments of 0-50°C.

The **NTR1000-P** is ideal for control, test, and measurement applications requiring a more stable and accurate clock than is provided with most DOS systems. The card features the National Semiconductor D8570A chip with battery-backup power and is referenced to a stable crystal with a known aging rate. To further enhance precision and accuracy, the oscillator is calibrated to the normal operating temperature of the PC for the best reference possible.

Other features of the card include 0.01 second accuracy, user-definable automatic update intervals from 1 minute to 45% days, extended AT interrupt support, a Novell driver, and a lifetime guarantee. The card is ideal for use with networks. By installing the card in the



network file server, any node on the network has access to highly accurate, stable time.

The **NTR1000-P** Real-time Clock card sells for \$199.00 and includes the board, manual, and software.

Industrial Computer Source  
10180 Scripps Ranch Blvd.  
San Diego, CA 92131-1298  
(800) 523-2320 • Fax: (619) 271-9713

#504

## SINGLE-CHIP FLASH DISK

Eurom has announced its patented **DiskOnChip** solid-state, nonvolatile memory chip with a capacity of up to 16 megabytes of unformatted "disk space" that fits pin-for-pin into a standard personal computer BIOS EPROM socket. The chip is targeted for hand-held and embedded computer designers seeking to add a bootable storage device, extend battery life, and maximize space for mobile systems.

The **DiskOnChip** is a **100 %** plug-and-play replacement for a system BIOS chip so that no additional software or

hardware reconfiguration is required. The chip uses the embedded Flash File System to emulate the organization and management of a standard generic read/write disk device.

**DiskOnChip** provides the user with a BIOS and a 1-16-MB (unformatted) solid-state disk in one-chip package. By adding a secondary storage device, it frees up a PCMCIA slot for essential communication modules used with portables and hand-held computers. The flash memory operates on only 5 volts, provides extremely low battery consumption compared with a hard drive, and requires no battery backup or maintenance.

By using a data transfer

rate of about 50M bits per second, the chip boosts productivity by making system boot-up time 10 to 20 times faster than system boot-up from a standard mechanical hard drive.

**DiskOnChip** also features interleaving support for fast write and erase operations.

**DiskOnChip's** features are possible by using an ASIC chip and circuitry to support flash memory programming as well as memory and logic blocks. The logic block communicates between the system and the memory block, intercepts the disk boot routine, and intercepts BIOS calls during flash programming. The memory block contains a minimum of one flash memory chip which

contains the system BIOS, a Flash File System, socket, and PCMCIA services, with the remaining flash memory as a formatted disk.

The 1 MB version of the **DiskOnChip** is offered for evaluation at \$99 per chip in quantities up to 100. Two- and four-megabyte versions of the chip are also available.

Eurom, Inc.  
556 Mowry Ave., Ste. 103  
Fremont, CA 94536  
(510) 505-9083  
Fax: (510) 505-9084

#505



# NEW PRODUCT NEWS

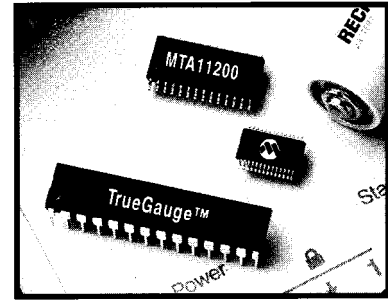
## BATTERY MONITOR AND CHARGE CONTROL

A low-power CMOS integrated circuit that provides accurate battery monitoring and charging control has been introduced by the ASSP Division of Microchip Technology. The **TrueGauge MTA11200** helps predict imminent battery failure and provides precise real-time battery capacity measurement and charge control.

TrueGauge is designed to operate with either nickel-cadmium, nickel-metal hydride, or lead-acid battery packs. It is ideally suited for use in portable computers, portable video equipment, cellular phones, and other devices that rely on rechargeable battery technology. TrueGauge resides in the battery pack where it continuously monitors battery condition. It calibrates itself automatically and continuously, maintaining a high level of accuracy, and excels in applications where a battery gauge is required to prevent an interruption in use or data loss due to insufficient battery power.

TrueGauge measures the battery's total capacity during automatic calibration and factors the data into the state-of-charge calculation to determine the remaining battery capacity. To extend battery life, TrueGauge

requests conditioning cycles at regular intervals based on battery usage, and can output battery parameters through a simple one-wire transmission link. Parameters include remaining capacity, total capacity, voltage, current, and temperature. The product has been specially designed to avoid errors due to noise, variations in load current, and deep discharge situations.



A development system that allows system designers to design and implement full production battery fuel gauge products for specific applications is also available. TrueGauge sells for \$3.75 in quantities of 10,000. The development kit sells for \$499.00.

Microchip Technology, Inc.

2355 West Chandler Blvd. • Chandler, AZ 85224-6199  
(602) 786-7200 • Fax: (602) 899-9210

#506

## ► Project Part 5 ◀

### New Stuff

TDA8444 I <sup>2</sup> C bus octal 6-bit DAC	9.30
MT8888 DJMF Transceiver (Intel bus interface)	10.50
PCF8583 I <sup>2</sup> C bus clock/calendar/counter w/ 256 byte RAM	11.70
UDN2989W Dual full-bridge stepper driver: 50 V, 2 A	13.80
PCF8591 I <sup>2</sup> C bus quad 8-bit ADC, single 8-bit DAC	15.10
BQ2003 Fast-charge NiCd/NiMH battery controller	14.55
MT8808 8x8 bipolar signal analog crosspoint	16.10
BQ2001 Battery Energy Management Unit	22.75
HBCS-1100 HP high-resolution bar code sensor	32.80
CH1840 FCC pre-approved DAA (full function, street legal phone hookup)	72.50

### Firmware Furnace Embedded '386SX Stuff

Complete schematics (15 pages or so)	5.00
DS2400 Silicon Serial Number (2 chips)	8.40
MAX691 Power supervisor	11.85
Firmware Development Board ICs & parts (through Issue 36)	45.70

### Firmware Flyers (prices are postpaid in US / with any parts order)

8051 Firmware Debugging Techniques (65 pages w/ 3.5" diskette)	18 / 15.00
Introduction to the 8051 Instruction Set (46 pages)	10 / 8.00
8051 Instruction Reference Card	3 / 2.00
8255 Cheat Sheet Reference Card	3 / 2.00

### IR Remote Control

LD273 dual IR LED (bright, wide beam)	2.10
IR3C02A laser diode controller (±5, for LT022MC/LN9705P laser)	2.30
IR3C07 laser diode controller (+5V, for LT032PD/LN9705 laser)	2.85
PH302 fast IR photodiode	3.10
GP1U52Y 40 kHz IR receiver (side-looking)	4.00
IS1U60 38 kHz IR receiver	4.80
IRSAMPLE parts (PH302, LM311, etc. See MCIR-Link article, INK 29)	5.70
Excellent IR filter (opaque to visible light, 35 mm slide mount)	6.75
MC145030 IR encoder/decoder	6.75
IR I/O: IS1U60, LD273, CD4047, 2N2907, red LED, schematic (IR-Link!)	10.40

### Stepper Motor & Power Drivers

ULN3751Z Power op amp (±3V to ±13V supply, 3.5 A output)	4.60
UDN2983B Dual full H-bridge bipolar stepper motor driver	5.45
UCN5841A Serial-input, 8 latched 500 mA sink drivers	6.90

...and much more...

JPS Ground/2nd day \$6.519 to 48 US states, COD add \$4.50. PO Boxes and Canadian orders \$6 for USPS! mail Check, MO, or COD only. no credit cards POs add \$50, call first. NC residents add 6% sales tax Quantity discounts start at five parts. Data sheets included with all parts.

Call/write/fax for seriously tempting catalog...

**Pure Unobtainium**

► Your unusual part5 source ◀

13109 Old Creedmoor Road • Raleigh, NC 27613

FAX/voice (919) 676-4525

## Build your own Neural Net! - and Expert System!

### EASY NEURAL NETWORKS



Easiest way to quickly learn about this fascinating new technology - includes a working Neural Network you can train! \$59 \*\*\*\*

### EASY EXPERT SYSTEMS

Build your own Expert System in about one day! Includes 40-Page Knowledge Engineering Manual, Quick Ref Card for quick start, and powerful Stock Market Expert System! \$69 \*\*\*\*

**BOTH \$99 + Includes 3 BIG Catalogs and FREE Talking Expert System Demo + Order now and receive a free copy of talking PC Therapist - usually \$64.95**

3 Catalogs & Demo Disk - Send your check for \$5 We always pay postage

Please specify disk size or we ship 3.5" - Check, American Express, or P.O. to: **THINKING SOFTWARE, INC.**

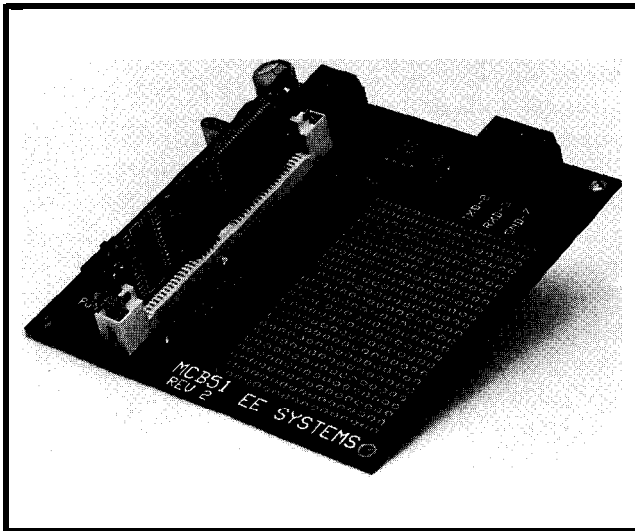
46-16 65TH PLACE - Dept cc201

WOODSIDE, N.Y. 11377 PHONE (718) 803-3638

# NEW PRODUCT NEWS

## SINGLE-BOARD CONTROLLER

EE Systems has announced its new MCB51 Single-board Controller. This easy-to-program 8051-compatible microcontroller board incorporates the Dallas Semicon-



ductor DS2250T controller, which is the SIP Stik version of the DS5000T DIP package.

All program and data memory is contained in 32K bytes of high-speed nonvolatile CMOS RAM. There are no EPROMs to remove and program. Downloading applications into the board is accomplished by connecting the board to a DOS PC serial port. An on-chip serial loader ROM along with a specialized DOS program (included with the package) makes this downloading process very quick and easy.

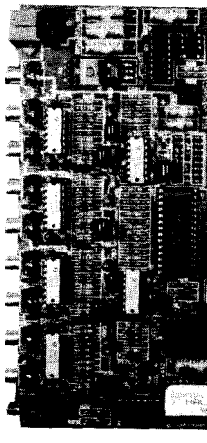
The 3.75" x 3.75" board has all of the standard 8051 features plus 30 I/O lines, a watchdog timer, real-time clock/calendar, power fail interrupt, and crash-proof power-down management. A prototyping area is also included on the board. A free BASIC compiler/assembler, with several program examples, is also included.

The MCB51 sells for \$149.95. A complete C development system for the MCB51 is also available for \$99.95.

EE Systems  
50935 Hill Dr.  
Elkhart, IN 46514

#507

# CIRCUIT CELLAR KITS



## HAL-4

EEG Biofeedback Brainwave Analyzer

The HAL-4 kit is a complete battery-operated I-channel electroencephalograph (EEG) which measures a mere 6"x7". HAL is sensitive enough to even distinguish different conscious states—between concentrated mental activity and pleasant daydreaming. HAL gathers all relevant alpha, beta, and theta brainwave signals within the range of 4-20 Hz and presents it in a serial digitized format that can be easily recorded or analyzed.

HAL's operation is straightforward. It samples four channels of analog brainwave data 64 times per second and transmits this digitized data serially to a PC at 4800 bps. There, using a Fast Fourier Transform to determine frequency, amplitude, and phase components, the results are graphically displayed in real time for each side of the brain.

HAL-4 kit..... \$179.00 plus shipping

• The Circuit Cellar Hemispheric Activation Level detector is presented as an engineering example of the design techniques used in acquiring brainwave signals. This Hemispheric Activation Level detector is not a medically approved device, no medical claims are made for this device, and it should not be used for medical diagnostic purposes. Furthermore, safe use requires that HAL be battery operated only!

Sonar Ranging Experimenter's Kit  
Targeting ♦ Ranging ♦ Machine Vision

The Circuit Cellar TI01 Ultrasonic Sonar Ranger is based on the sonar ranging circuitry from the Polaroid SX-70 camera system. The TI01 and the original SX-70 have similar performance but the TI01 Sonar Ranger requires far less support circuitry and interface hardware.

The TI01 ranging kit consists of a Polaroid 50-kHz, 300-V electrostatic transducer and ultrasonic ranging electronics board made by Texas Instruments. Sonar Ranger measures ranges of 1.2 inches to 35 feet, has a TTL output when operated on 5V, and easily connects to a parallel printer port.

TI01 Sonar Ranger kit.....\$79.00 plus shipping

## CHECK OUT THE NEW CIRCUIT CELLAR HOME CONTROL SYSTEM

- ♦ Expandable Network
- ♦ Digital and Analog I/O
- ♦ X-10 Interface
- ♦ Voice
- ♦ IR Interface
- ♦ Remote Displays
- ♦ Telephone Interface

**Call and ask about the MCS II**

To order the products shown or to receive a catalog, call: **(203) 875-2751** or fax: **(203) 872-2204**

Circuit Cellar Kits • 4 Park Street • Suite 12 • Vernon, CT 06066

#106

# NEW PRODUCT NEWS

## DEVELOPMENT TOOLS DIRECTORY

The New **Motorola Microcontroller Development Tools Directory**, published by MW Media, lists more than 40 suppliers of hundreds of development tool products that support designs using the Motorola M68HC05, 'HC08, 'HC11, and '300 families.

Charts break down the Motorola microcontroller family by part numbers and include ROM, RAM, timers, packaging, and additional comments detailing each chip.

The directory lists all key suppliers of development tools and products. The 126-page book is composed of product data sheets and line cards for each product. Each page describes hardware and software development tools such as in-circuit emulators, programmers, evaluation boards, debuggers, compilers, and full development systems. The publication is free, but there is a \$5.00 shipping charge for shipments in the U.S. and an \$8.00 shipping charge for international orders.

MW Media  
50 W. San Fernando, Ste. 675 • San Jose, CA 95113  
(408) 286-4200 • Fax: (408) 288-4728

#508

## REMOTE CONTROL OF COMPUTER PERIPHERALS

**ZapLink**, an infrared wireless data link introduced by Control Bits Corporation, uses a computer RS-232 serial port to provide wireless communication between computers and control of peripherals. Specially configured for laptop computers, ZapLink provides an infrared transceiver which attaches to the serial port of each computer and peripheral to be linked.

An infrared signal carries data between the devices, provided they are within line of sight. The company also offers devices which enable the infrared signal to be transmitted into other rooms and around corners.

ZapLink is available in several configurations. Its transceiver can be mounted on an antenna, a telescoping antenna, or be attached to a wire and mounted to a nearby surface. ZapLink operates at up to 9600 bps and requires 12 VDC. A 115-VAC adapter is available.

Control Bits Corp.  
8429 Center Dr. N.E. • Minneapolis, MN 55432  
(612) 783-2157 • Fax: (612) 783-2160

#509

The  
*only*  
8051/52  
BASIC  
compiler  
that is  
100 %  
BASIC 52  
Compatible  
*and*  
has full  
floating  
point,  
integer,  
byte & bit  
variables.

- Memory mapped variables
- In-line assembly language option
- Compile time switch to select 8051/8031 or 8052/8032 CPUs
- Compatible with any RAM or ROM memory mapping
- Runs up to 50 times faster than the MCS BASIC-52 interpreter.
- Includes Binary Technology's SXA51 cross-assembler & hex file manip.util.
- Extensive documentation
- Tutorial included
- Runs on IBM-PC/XT or compatible
- Compatible with all 8051 variants
- **BXC51** \$295.

508-369-9556  
FAX 508-369-9549



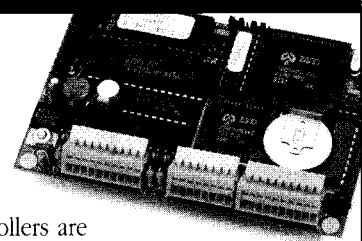
Binary Technology, Inc.  
P.O. Box 541 • Carlisle, MA 01741



#107

## LOW COST MINIATURE CONTROLLERS

Save time.  
Save money.



Our wide range of C-programmable miniature controllers are ideal for control applications, data acquisition, and test and measurement. Compact and low in price (the Little PLC™ above is 2"x3" and \$195), these controllers are programmed with our easy-to-use Dynamic C™ development system. Our controllers feature digital I/O, ADCs and DACs, relays and solenoid drivers, RS232/RS485 serial ports, battery-backed memory and time/date clock, LCDs, keypads, enclosures and more!



1724 Picasso Ave.  
Davis, CA 95616  
916.757.3737  
916.753.5141 FAX

24-Hour AutoFAX 916.753.0618. Call from your FAX. Request catalog #18.

08

## FEATURES

14

Programmable Lightware

26

In the Eye of the Beholder

36

interrupt-free Design

40

An Assembly Language Programming Aid

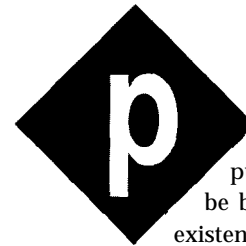
## FEATURE ARTICLE

John Hait

# Programmable Lightware

## How to Develop Photonic Products for Fun and Profit

You think those new processors such as the Pentium and Alpha are fast? Ever consider a light-based computer? It may not be long before you SEE one of these.



Photonic computer hardware will be brought into existence when more computer-generated graphics (whether they be plotted, printed out, or photographically reproduced) are assembled with off-the-shelf laser systems, and sold.

What is a *photonic computer*? It is a computer made up of *photonic transistors*. A photonic transistor is essentially a tiny photograph that is able to perform Boolean logic and photonic signal amplification. In a photonic computer, photographs replace silicon, and laser light replaces electricity.

Computer-generated graphics produced by applying the laws of optics can be used to create synthetic holograms. Transparencies reduced directly from synthetic holograms can control light beams just as natural laws predict. Synthetic holograms calculated to direct light in a photonic transistor configuration can be used as "schematics" to produce real holograms that perform logic and amplification of "digital light beams" to form the equivalent of processors, ROM, RAM, I/O, and so forth.

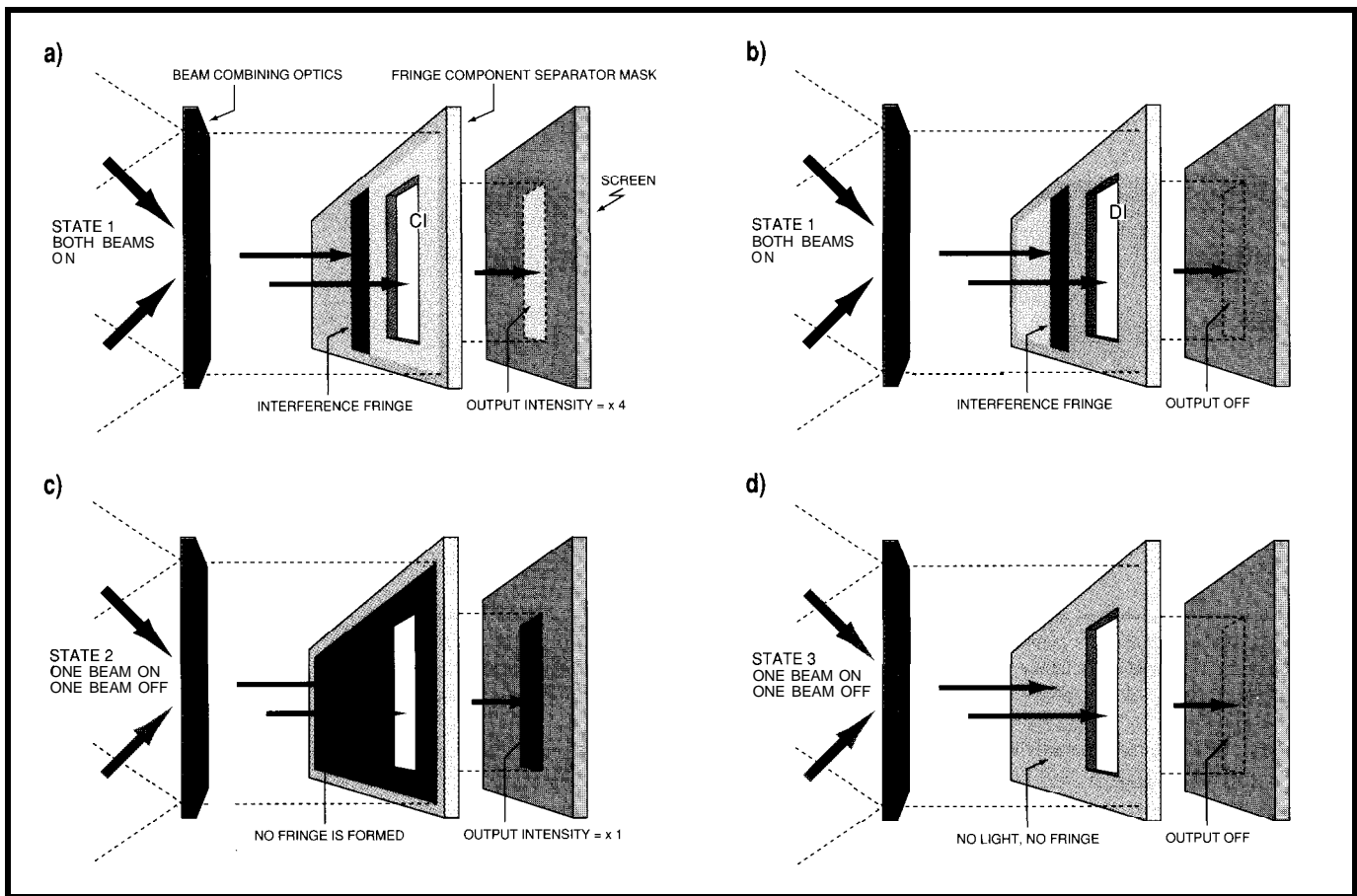


Figure 1-A simplified photonic transistor showing an amplifier/boolean OR circuit and an XOR circuit. When two laser beams are combined (a & b), they form an interference fringe onto a fringe component separating mask. If one beam is shut off (c), the fringe goes away and light goes through the hole in the mask to produce an "on" output of intensity 1. If both beams are shut off (d), the output is off. If the hole in the mask is placed at the position of constructive interference (CI), the output through the hole is four times the intensity of a single beam. The device will amplify a modulated input signal if the other beam is kept on all the time, or it will function as a boolean OR circuit if both beams are modulated. If the hole is placed at a position where destructive interference (DI) occurs (b), then a boolean XOR circuit is produced.

A great variety of interconnecting holograms will be required in a complete photonic computer, because the exact holograms needed for each operation varies from component to component. Yet, before fully photonic computers reach the market, myriad discrete photonic components, each constructed from a few holograms, will be developed and sold... just as was the case with early electronic products.

Since the holographic components needed to build photonic transistor-based products are produced by software, photonic computers and other devices based on the photonic transistor are essentially massively parallel programmable hardware systems-programmable lightware! This article explains how to produce synthetic holograms, as well as the software related to synthetic hologram production, for fun and profit.

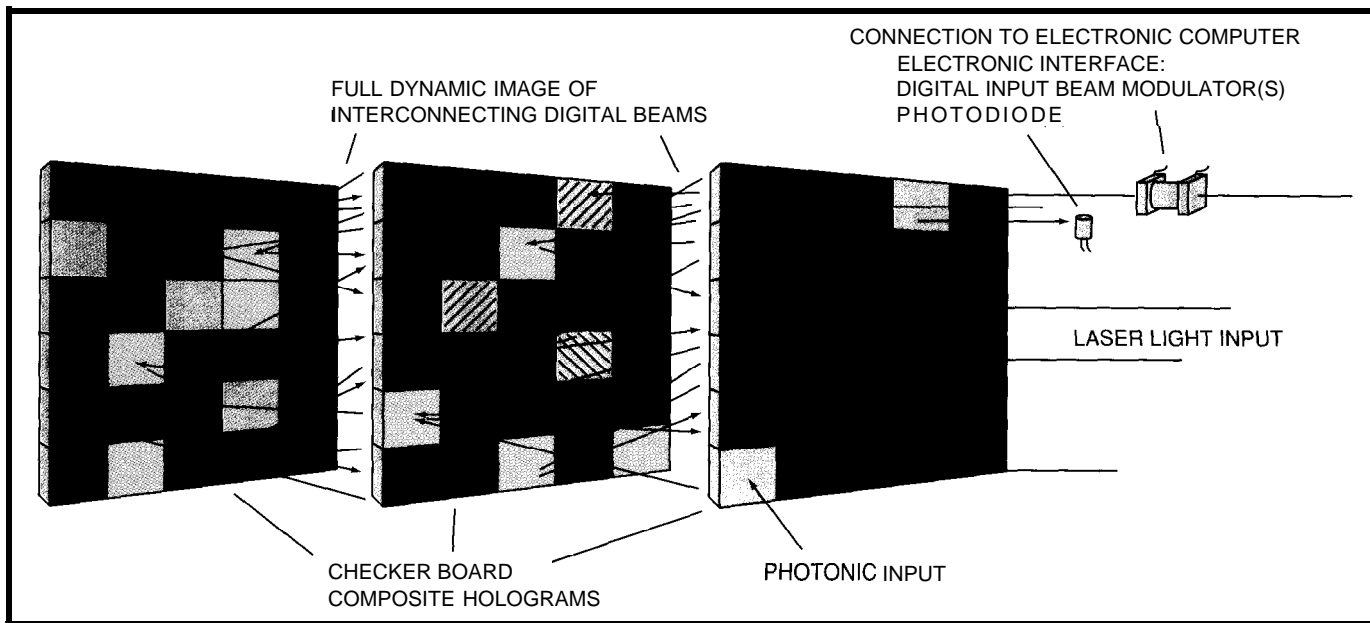
## WHY BUILD A PHOTONIC COMPUTER?

Two reasons: speed and money. Information can be impressed onto light rays through a variety of techniques. The intelligence-carrying photons don't slow down as they pass through a photonic transistor the way electrons do in semiconductors. Thus, information is able to be processed at a higher speed, using the widest possible bandwidth.

By way of comparison, a '486 running at 66 MHz takes 15 nanoseconds per internal clock cycle, and the latest, hottest, Cray 3 supercomputer uses 2 ns per cycle. In 2 ns, light travels 60 cm (23 inches) and in 15 ns, it goes 450 cm (14 feet 9 inches). You can fit a whole lot of photonic transistors in 60 cm, let alone 14 feet! So, a '486 emulated in light, even using oversized test transistors, would run faster than a Cray 3.

Naturally, it's not expected to be very long before photonic transistor equipment begins to shrink. As the size goes down, speed goes up. Already, 3M is able to put 20,000 lenses in one square centimeter. Photonic transistors that small will easily outpace the theoretical limits of electrons.

Before then, macro-sized devices must be produced and tested. Organizational layouts of photonic transistor arrays have to be designed and optimized. The many details for producing the massively parallel programmable synthetic holograms have to be worked out. And methods for making registers, processors, memories, ALUs, and all of the other structures that make up computers have to be derived. Once these steps are complete, the macro hardware can be scaled down mathematically to produce full-speed, optical supercomputers.



**Figure 2**—A holographic computer is made up of standardized little hologram squares that are combined to make checkerboard composite holograms. Each square is part of a photonic transistor. Laser light directed through the holograms produces dynamic images, where the beams from each square blink on and off as digital signals carrying data to the text little holographic photonic transistor. Any electronic computer architecture may be duplicated in this way, as a superspeed optical computer.

This development process has the distinct advantage that much of the work can be accomplished using conventional equipment. Synthetic holograms sufficient for producing research and demonstration components can be produced on ordinary computers and printed on VGA or SVGA screens and laser printers. These "printouts" are then photographically reduced to become actual functioning devices. The resulting holograms can be mounted in a unit with an off-the-shelf laser to make functioning photonic products.

The hologram-generating programs and even the simplest working photonic products will become very valuable to hardware hackers and entrepreneurs involved in photonics. Educational products are needed so that developers will be able to keep abreast of the latest technology. By using the equipment already available, programmable lightware provides a physical means for the production of actual saleable products.

Yet, there is another very important advantage. Programs which produce holograms can also be used to test the *holographic components*. Each potential photonic product can be simulated to see how it will function without having to construct a single actual component. Thus, photonics

products can be created and tested without buying a laser or taking even a single photograph.

Why would you be interested in producing synthetic holograms on your own computer? Today, we are on the ground floor of a new and exciting technology. Already, long-distance telephone companies are investing millions of dollars to switch from wires to fiber optics, trading their electrons for photons, so to speak. Every experimenter in the world would like to buy the world's fastest components if they were affordable. Computers made out of photographs will be less expensive to build than ones made out of semiconductors. So, photonic computers can be expected to outperform and outsell even the best 35-million-dollar supercomputers.

There is another very important reason. We are instituting a "Distributed Development and Sales Network," whereby individuals from all over will be given the opportunity to share in the development of photonic components and related software for sale through our network.

### LOGICAL LIGHTWARE

A detailed discussion of photonic transistor basics is contained in the August 1992 issue of the *Computer Applications Journal*, so I will not

duplicate that information here. The basic concepts needed for this discussion are shown in Figures 1a and 1b. Two laser beams are brought together to produce an interference fringe. The fringe is composed of dark areas of destructive interference (DI), and bright areas of constructive interference (CI). Photons that ordinarily would have gone straight towards the DI area are diverted into the CI area by the action of interference. Thus, the CI areas are four times brighter than a single beam by itself.

How can that be? Let's compare Figures 1a and 1c. A single beam floods the entire area occupied by the fringe when both beams are on (Figure 1c). For each photon that lands in the area that corresponds to the CI area of Figure 1a, another lands in the area that corresponds to the DI area. Turning on another beam of equal strength (Figure 1a) doubles the number of available photons. So now we are looking at four photons. Two headed for the CI area, and two headed for the DI area. The action of interference, though, diverts the photons headed for the DI area into the CI area. Thus, where there was one photon in the CI area with a single beam (Figure 1c), now there are four (Figure 1a). This property is an experimentally verifiable fact of physics.

If the component parts of the fringe are separated using a mask, or strategically placed holograms, several functions emerge. Separating the fringe component parts creates a differential output between the single beam state (Figure 1c) and the double beam state (Figure 1a).

In the example above, the single beam has two photons and the double beam has four. Taking the entire combined beam into account, the differential between having one beam on and having two beams on is only double the intensity. Adding the mask drastically alters that output differential.

With a fringe component separator that selects only the CI area(s), the single beam output is cut in half (Figure 1c), whereas the double beam output remains the same (Figure 1a). If both beams are off (Figure 1d), the output is zero. So, the output switches from zero photons to one to four as each beam is turned on. By selecting different parts from the fringe, different computer functions can be produced.

In Boolean terms, the device is an OR gate because it produces an "on" output if either or both input beams are on. In order to be off, both input beams must be off. The problem is that without the mask, light levels (in the "on" state) fluctuate from one to two times a single beam intensity, and with the mask, they fluctuate from one to four times a single beam intensity. Fluctuations like that may cause complications in subsequent devices.

In actual beams, there are many photons, and not all of them are precisely located in the center of the CI area. If the hole in the mask is located so as to select photons partially from a CI area and partially from a DI area, a balance can be found where the output intensity when both beams are "on" equals the single beam state. The result is the OR function that can be interconnected directly with other devices.

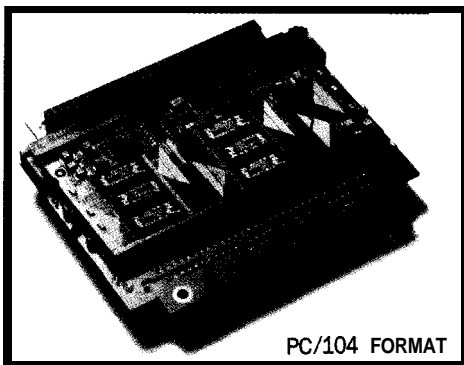
If the mask separates light from a DI area (Figure 1b), an exclusive-OR (XOR) device is made. The output

differential switches from zero (Figure 1d) to one (Figure 1c) and back to zero (Figure 1c). Since only two functions are needed to produce everything that is done on the motherboard of an electronic computer, every computation can now be done with light.

If we note the process carefully, the actual power, or total amount of light is reduced in the process. That shouldn't surprise us. Put your finger on a '486 and consider how much energy is wasted as heat. Images generally require much less energy than a '486, so we don't need high-powered lasers. But, light levels will need to be matched at each stage. So, an amplification function will be needed somewhere along the line in order to be able to interconnect all of the components properly.

However, many light speed functions are able to be accomplished before the on/off contrast reduces to the point where information is lost. For example, consider the address decoding function for a hypothetical photonic RAM. The entire address

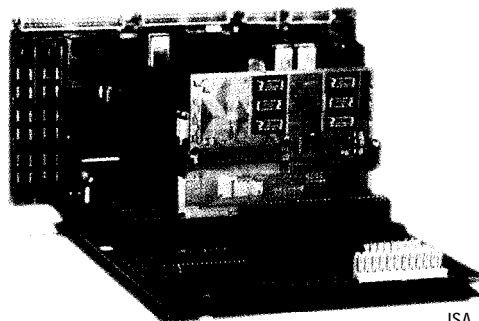
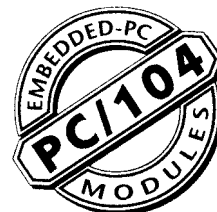
## 25 MByte **FlasHD** Memory Module



PC/104 FORMAT

- ▶ Hard Drive replacement; supports M-Systems
- ▶ Nonvolatile Flash™ memory board
- ▶ Self Boot mechanism and PC auto detection
- ▶ Available with up to 25MBytes of storage
- ▶ Read/write memory system
- ▶ "Boot Block" of 256KBytes allows DOS in ROM
- ▶ Byte writable, 64KByte sector erasable
- ▶ On-board DC to DC convertor for single +5 volt operation

*"The FlasHD card allows a system to boot without mechanical drives, execute applications and write up to 25MBytes of non-volatile storage."*



ISA FORMAT

For a list of our international distributors, please contact our head office at:

*"For the OEM with Embedded PC Applications"*

**megatel**

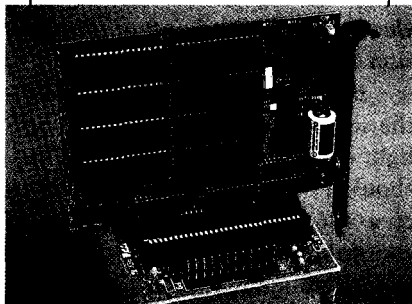
**(416) 245-2953**

**megatel computer corp.**

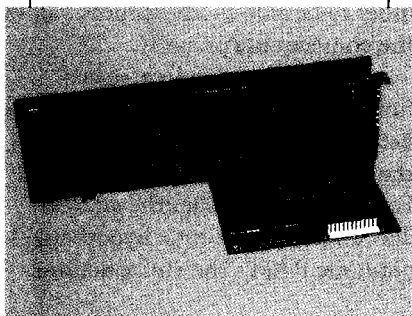
125 Wendell Ave.  
Weston, Ont. M9N 3K9 Canada  
Fax: (416) 245-6505

FlasHD is a trademark of Intel Corp. and Megatel is a registered trademark of Megatel Computer Corp.

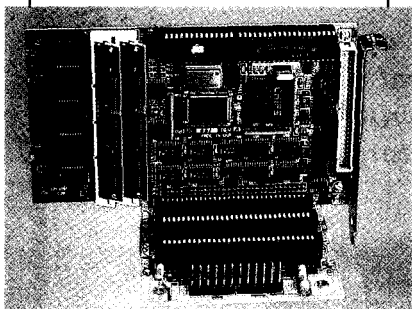
**VMAX**  
**QUALITY PRODUCTS**  
**RESPONSIVE SERVICE**  
**RELIABLE DELIVERY**



**SOLID STATE DISK — \$135\***  
 1/2 Card 2 Disk Emulator  
 EPROM, FLASH and/or SRAM  
 Program/Erase FLASH On-Board  
 1M Total, Either Drive Bootable



**25MHZ 386DX CPU — \$695\***  
 Compact AT/Bus or Stand Alone  
 (In-Board SVGA, IDE, FDC, 2 Ser/Bi-Par  
 FLASH/SRAM Drives to 2.5M  
 Cache to 128K, DRAM to 48M



**TURBO XT**  
**w/FLASH DISK — \$266\***  
 To 2 FLASH Drives, 1M Total  
 DRAM to 2M  
 Pgm/Erase FLASH On-Board  
 CMOS Surface Mount, 4.2" x 6.7"  
 2 Ser/1 Par, Watchdog Timer

All Tempustech VMAX products are  
 PC Bus Compatible. Made in the  
 U.S.A., 30 Day Money Back Guarantee  
 \*QTY 1, Qty breaks start at 5 pieces.

**TEMPUSTECH, INC.**

**TEL: (800) 634-0701**

**FAX: (813) 643-4981**

Fax for 295 Airport Road  
 iast response! Naples, FL 33942

decoding and information read or write operation is able to take place at the speed of light, even with each stage in the process operating at a diminished power level from the previous stage. Only on the read function, after the RAM has been read, must the data bus be amplified in order to bring it back up the high levels useful to the processor.

We have a precedence for this in electronics. Early computers, and the internal workings of many computer chips, use diode-resistor logic for decoding addresses because they take up so little space. The output is then amplified back to TTL levels after the multistage logic has been performed.

While complete processes may be created using a descending power arrangement, amplification will be needed in order to restore the proper contrast ratio. There are light amplifiers available on the market, albeit slow and expensive ones. These could be used.. but why, when we have the photonic transistor amplifier?

If the mask separates the CI area (Figure 1a), and one beam is kept on all the time, the information on the modulated beams enters the transistor, switching from zero intensity (Figure 1c) to one (Figure 1a), for a contrast ratio of one. But, it exits switching from one to four, for an intensity differential (contrast ratio) of three. Thus the modulated signal has been amplified by increasing the contrast between off and on.

In a comparison to an electronic amplifier, the beam that remains on, is like the DC current from the battery. Without a power supply, the amplifier quits amplifying. A small modulated input signal switches the transistor on and off. The modulated output signal is larger than the modulated input signal. It is amplified: its "contrast ratio" has been increased. Where did the modulated output signal get the extra energy! Why, from the battery of course.

Likewise in photonics, the constant beam supplies energy to the modulated beam so that the modulated output beam has more energy than the modulated input beam, even though the overall process is less than

100% efficient. The same is true of an electronic transistor.

Many optical beam combining methods, including some holograms, block out half of the light from the input beams. In these cases, power amplification may be negated, even though the contrast ratio is amplified. The increase in contrast will be sufficient for many applications, but not for others.

Some of the methods for optimizing beam-combining optics, so as to minimize light loss, include the use of phase holograms and reflecting holograms such as the ones put on credit cards. These can be used to combine beams so as to cause an actual increase in total light power. As illustrated above, the amplified output beam has four photons for every two photons in the original modulated beam for a power amplification factor of two.

Simple photonic transistor amplification produces a better contrast ratio, but the output level remains on at a low level during the off state. To reduce this constant bias, the ratio of CI area (that passes through the mask) to DI area (which is blocked by the mask) is adjusted so as to make the CI area as small as possible and the DI area as large as possible. Obviously, the beam-combining optics must create a matching fringe with a wide DI area and a narrow CI area. That's where holograms come in.

## HOLOGRAPHIC PHOTONIC TRANSISTORS

Holograms are able to produce complex images that can be designed to produce complex fringes with predetermined qualities. For example, a simple zone plate hologram can increase the contrast ratio by 400 times by focusing the light onto a small dot much as a magnifying glass does. When designed to accept single- and double-beam inputs, the contrast ratio can really become quite large.

Holograms are especially useful for producing differential photonic amplifiers. A differential amplifier is similar to a simple amplifier except



that a fringe is produced by the constant bias beam having its CI image completely blocked by the mask.

When the second beam comes on, the combined image is calculated so as to relocate the photons into a second image that is located elsewhere on the mask...and that's where we put the hole. From another viewpoint, the constant beam produces a fringe with its CI area blocked by the mask. When the second beam comes on, the CI area of the combined fringe is moved over to the location of the hole.

The result is the output contrast switches from all the way off to many times the level of a single modulated beam. If phase holograms are used, the power in the modulated output beam can be nearly twice the power of the modulated input beam. Cascading many such amplifiers together can then be used to raise both the power and contrast ratio to any needed level.

## MAKING A HOLOGRAPHIC COMPUTER

Figure 2 shows one possible configuration for building a holographic supercomputer. It consists of a large number of holograms laid out in a checkerboard fashion, each square being either a hologram, a fringe component separator, or both.

Light from each square is directed by its hologram to the squares in the next plane. In this way, the logic devices may be both constructed and interconnected. Even data input and output beams can be produced, and laser "power" may be distributed to each of the components as needed. By arranging the various little square holograms so as to cooperate together, complete computer architectures may be constructed.

Holograms also have the ability to produce very complex images that can be selected by individually directing, not plain laser light at the hologram, but by projecting an image onto the succeeding hologram.

In Figure 2, the composite of little holograms at each plane can be considered as one big hologram with many subhologram parts. If an image is projected onto a composite hologram, some of the little portions will

be lit and some will not. By specially calculating each little hologram in concert with the others, a different image may be produced for each of a number of input images. How would that be useful?

If the input image consisted of a number of beams that represented an address bus, the combined output through an array of mask openings can be the completely decoded address. Thus for each combination of input beams projected onto the beam-combining hologram, only one mask hole will be lit. Or if negative logic is used (which has some advantages), all would be lit except one. If the output of such an address decoding array is directed to an array of photonic transistor flip-flops, the device is a RAM. If the output is directed to a viewing screen, the device is a completely photonic, high-speed video display! By the way, the right configuration can display in 3-D, too.

Are you ready to make some synthetic holograms?

## MAKING AND TESTING SYNTHETIC HOLOGRAMS

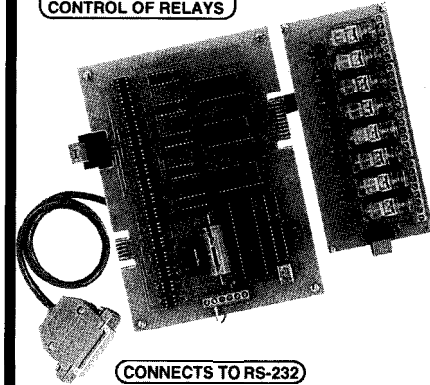
Synthetic holograms are computer-generated graphics. The programs are relatively simple, but they take a very long time to run. They are called *ray tracing* programs.

Figure 3 shows a standardized configuration that includes two standard input squares that cooperate together to produce a standard output. Each square may be a hologram, a mask, or a combination. Light from each input square is calculated using the wave nature of light so as to determine what pattern will be projected onto the output square. By doing two calculation sets, one for each input square, three different images can be produced on the same output square. One for when one beam is on by itself, and a second for when the other beam is on by itself. By summing these two calculated outputs, the result from having both beams on at the same time can be produced.

By writing a set of basic interchangeable table-oriented algorithms, several things become possible:

# RELAY INTERFACE

PROVIDES SOFTWARE CONTROL OF RELAYS

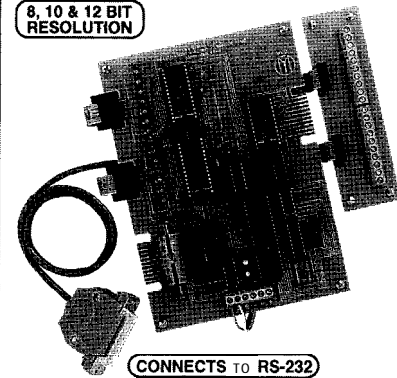


CONNECTS TO RS-232

**AR-16 RELAY INTERFACE** (16 channel).....\$ 89.95  
Two 8 channel (TTL level) outputs are provided for connection to relay cards or other devices (expandable to 128 relays using EX-16 expansion cards). A variety of relays cards and relays are stocked. Call for more info.  
**AR-2 RELAY INTERFACE** (2 relays, 10 amp)....\$ 44.95  
**RD-8 REED RELAY CARD** (8 relays, 10 VA).....\$49.95  
**RH-8 RELAY CARD** (10 amp SPOT, 277 VAC)....\$ 69.95

# ANALOG TO DIGITAL

8, 10 & 12 BIT RESOLUTION



CONNECTS TO RS-232

**ADC-16 A/D CONVERTER\*** (16 channel/8 bit)....\$ 99.95  
**ADC-8G A/D CONVERTER\*** (8 channel/10 bit)....\$124.95  
\*Input voltage, amperage, pressure, energy usage, joysticks and a wide variety of other types of analog signals. RS-422/RS-485 available (lengths to 4,000').  
Call for info on other A/D configurations and 12 bit converters (terminal block and cable sold separately).  
**DC-8E TEMPERATURE INTERFACE\*** (8 ch)....\$ 139.95  
Includes term. block & 8 temp. sensors (-40° to 146° F).  
**TA-8 DIGITAL INTERFACE\*** (8 channel).....\$ 99.95  
Input on/off status of relays, switches, HVAC equipment security devices, smoke detectors, and other devices.  
**TA-8D TOUCH TONE INTERFACE\***.....\$ 134.95  
Allows callers to select control functions from any phone.  
**S-4 PORT SELECTOR** (4 channels **RS-422**)....\$ 79.95  
Converts an RS-232 port into 4 selectable RS-422 ports.  
**O-485 (AS-232 to RS-422/RS-485 converter)**.....\$ 44.95

**EXPANDABLE**...expand your interface to control and monitor up to 512 relays, up to 576 digital inputs, up to 28 analog inputs or up to 128 temperature inputs using the PS-4, EX-16, ST-32 & AD-16 expansion cards.

**FULL TECHNICAL SUPPORT**...provided over the telephone by our staff Technical reference & disk including test software & programming examples in Basic, C and assembly are provided with each order.

**HIGH RELIABILITY**...engineered for continuous 24 hour industrial applications with 10 years of proven performance in the energy management field.

**CONNECTS TO RS-232, RS-422 or RS-485**...use with IBM and compatibles, Mac and most computers. All standard baud rates and protocols (50 to 19,200 baud).

Use our 800 number to order **FREE INFORMATION PACKET**. Technical information (614) 464-4470.

**24 HOUR ORDER LINE (800) 842-7714**  
Visa-Mastercard-American Express-COD

International & Domestic FAX (614) 4844856  
Use for information, technical support & orders.

**ELECTRONIC ENERGY CONTROL, INC.**  
380 South Fifth Street, Suite 604  
Columbus, Ohio 432155438

- \*Holograms can be produced to form specific output images.
- \*Images from each hologram and both holograms together may be simulated.
- \*Masks may be created to provide predetermined outputs given all input states.
- \*The operation of the full transistor can be simulated for any set of image inputs and for all states.
- Outputs from one transistor can be entered directly into a succeeding transistor.
- \*Outputs from one transistor can be routed to a succeeding device by creating a hologram that simply produces a plane output image, which angles over to the output position.

If viewed as a two-input, one-output device, each of the four possible states (including the moot state where both beams are off) can be simulated. If the two input squares are holograms and the output square is a mask, the operation of this synthetic photonic transistor may be simulated. Thus, your work may be tested without even having a laser or any other equipment other than a computer.

By using standardized, interchangeable, pixel amplitude tables for both input and output [albeit very large ones), holograms, masks, and the simulated operation of each can be produced from a single set of table-operating algorithms. The table arrangement permits complex wavefronts and images to be calculated from a standardized data set. Thus, the data output from one transistor can be entered as a light configuration into the next transistor. Complex products can be built up one transistor at a time.

Images calculated from pixel to pixel produce the same mathematical result regardless of which direction the light is traveling. Data output tables can be used as holograms from predetermined images, images from previous holograms, images from images, and masks from image combinations. Masks and holograms can be superimposed upon input

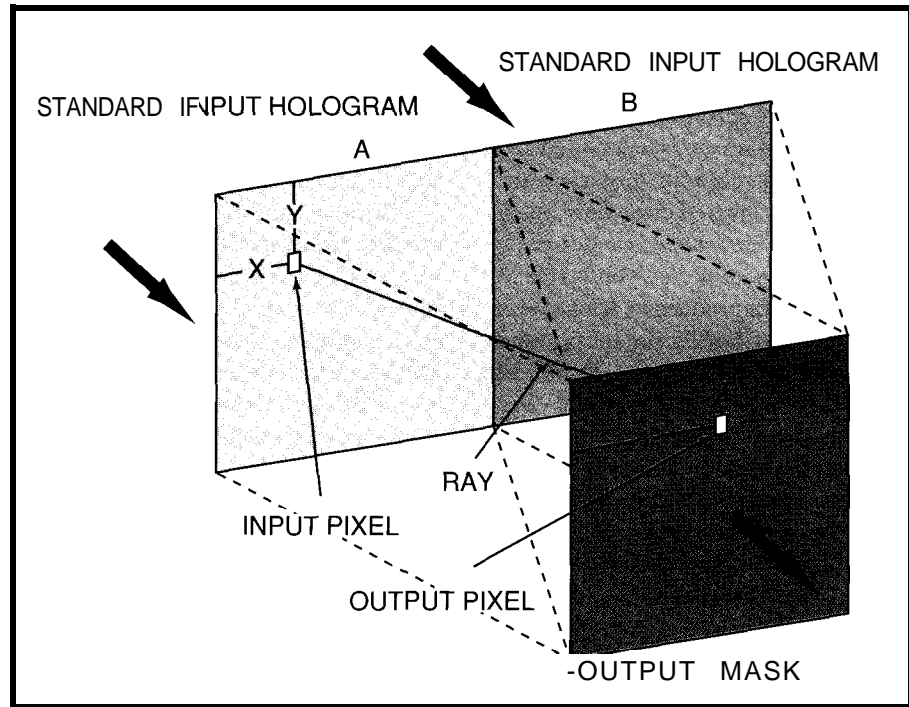


Figure 3-h a standardized photonic transistor, input holograms A & B provide two optical inputs to a transistor. The output mask separates the interference fringe images so as to produce signal amplification or boolean logic. Standardizing the hologram/mask configuration allows the outputs from each transistor to become the direct inputs to the next transistor. Thus, holograms and masks may be calculated easily and the operation of each transistor may be completely simulated and tested.

images so that each of the possible input states may be simulated.

### RAY TRACING PROGRAMS

Each standard square consists of an array of pixels. Light which exits each pixel initially heads toward all of the pixels of the output square. The line which connects an input pixel to an output pixel is a ray. Thus, every output pixel will have a ray coming toward it from every one of the input pixels that have a nonzero input amplitude.

By summing the instantaneous amplitudes of light that arrives at each pixel from every nonzero input pixel, the combined amount of light at that pixel, for that instant, may be determined. By calculating every ray from every input pixel to every output pixel, the entire output image may be produced.

The instantaneous amplitude from each ray is proportional to the phase of the light when it arrives at the output pixel. If that instantaneous amplitude is in phase with the input pixel (assuming a positive input amplitude), the output amplitude from that input

pixel is positive. If the ray produces an instantaneous amplitude that is out of phase with the input pixel, then its instantaneous amplitude is negative.

If one ray produces a +1 instantaneous intensity, and another also produces a +1, the result is +2 (constructive interference). If one produces a +1 and another a -1, the net result is zero (destructive interference). The calculated output for each pixel will be the algebraic sum of all instantaneous amplitudes from all input pixels.

What makes the phase change from one pixel ray to the next? Distance. The ray tracing program calculates the 3-D distance from each pixel in the input square to each pixel in the output square. The distance is converted to (a number + a fraction) of wavelengths. The wavelength fraction multiplied by the amplitude of the input pixel equals the instantaneous amplitude arriving at that output pixel.

A mask or hologram graphic can be superimposed onto a pixel table by setting all pixels that match corresponding black pixels in the graphic equal to zero.

## SYNTHETIC PHOTONIC TRANSISTOR DEVELOPMENT PROCEDURE:

First, you need to select an input table of pixel amplitudes either from a previous run or one with all pixels initialized to one (which represents a plane wave of equal amplitude directly from the laser). Next, calculate the image output from the A input side of the device. Now, produce a hologram by setting each pixel in the graphic to be either one or zero, to correspond with the output pixels values from the previous step. Set them by a predetermined range for the type of device in use (usually all negative values produce zeroed pixels).

Next you need to superimpose the graphic onto the original input table. Then you have to recalculate the image to determine if the desired output has been achieved, adjust the pixel setting parameters if needed, and perform a loop at this point until the desired output image is finally created.

Then you need to produce a similar hologram from the B side of the device. Now you can produce a combination image by summing the two previous output tables. At this point, it is a good idea to display and check all states for the production of the desired images. You may need to iterate the process until the optimum images are produced.

You will mask the output table (by zeroing the desired pixels) to produce the final transistor. Now you have to test all three "on" states using the same mask and the three previous output tables to ensure proper operation. In most cases, the result should be an image that is on or off (resulting in a single output table) rather than a complex set of images for each of the input possibilities. Otherwise things get real complex.

This photonic transistor is complete. Go on to the next device by adjusting the coordinates of the output table to become an input table. Repeat these steps until the product is done.

## EXAMPLE FLOWCHART PROGRAMS

I have included some examples to explain the process in the simplest possible way. They show how standardized tables can be used with various algorithms for producing a multitude of photonics products.

The actual table sizes, however, are much too large for these exact programs to function with practical working holograms. So use these programs more like flowcharts: as teaching aids for understanding the processes involved. Then, use your own skill to produce working programs in whatever language you wish.

Use the distance between the input squares and the output square as the minimum focal length times two, so that the output square size will be the same as the input squares. Note that tables should be calculated with at least 32-bit accuracy.

For a VGA screen, with 640 horizontal pixels, two 320x320 pixel

# FREE! HomeAutomation Catalog!

24 HOUR CATALOG REQUEST LINE  
Catalog Request by Fax 619-693-8892  
Customer Service Dept. 619-693-8887

# 1-800-422-4024

## Lowest Prices.



## Double Warranty.



## Best Selection.

*Lamp Modules \* Motion Detectors \* Keychain Remote Controls \* Drapery Controllers \* Software Development Kits \* Video Modulators \* Decorator Style Professional X-10 Equipment \* Wall Switch Modules \* HomeBase Intelligent Home Controller \* Universal Remote Controls \* Home Automation Plans \* Picture-in-Picture \* TeleMaster Telephone Interface \* Rain Sensors \* 900 MHz Cordless Phones \* Appliance Modules \* Wireless Security \* Custom Home Automator \* Video Distribution Equipment \* Home Automation Books \* Covert Video Cameras \* Wireless Wall Switches \* Business Phone Systems \* PC add-ons \* In-Wall PIRs \* Hacker's Bible \* Covert Video Cameras \* Smart Flood Lights \* Voice Control \* Professional Test Equipment \* Aquaphone \* Universal Module \* Video Sequencers \* X-10 Chime Module \* Motion Switch \* X-10 Compatible Stanley Garage Door Opener \* Panasonic Pocket Flip Phone \* Wireless Siren \* Whole-house Surge Protector \* Video Doorphone \* X-10 Thermostat Setback \* Caller ID \* Electric Doorstrike \* Special Module Powerstrips \* VoicePrint Phone \* And Lots Lots More!*

## Fastest Shipping.



ORDER TODAY...  
**HAVE IT TOMORROW!**  
UPS Next-Day & 2nd-Day Air Available!

## Fabulous Freebies.

Who says there's no such thing as a free lunch?  
There's not... but there are free lamp modules! Check out our new catalog for fabulous free gifts with every order!

# HOME CONTROL CONCEPTS

squares will display side by side. All calculations will be done in pixel-length units.

The flowchart-like routines shown in Listing 1 give a sense of how photonic transistors based on synthetic holograms are produced, but this list is by no means exhaustive. The problem is actual programs like this run at a snail's pace because the run time is multiplied by the number of input pixels that are not zero. However, there is a great deal of redundancy that one can capitalize on in order to speed things up.

Standardized squares produce standardized phase calculations. So, working programs will probably use table lookup procedures and precalculated constants that are not in the nested loop. Only the input amplitude at each pixel changes, so simple multiplication or addition need be the only math function within the loop.

Once certain types of transistors are produced, it may be easier to adjust the input light configuration to a

standard form so that preprocessed data can be reused as much as possible. In fact, the next logical step will be to produce CAD programs that use precalculated standardized units that can be combined into various products.

Additionally, this process is well suited to neural programming, Fourier transforms, and other methods for arriving at the iterated result much faster. The point is that actual working components can be synthesized within a standardized configuration much faster even though the computational methods may differ.

Obviously, early programs will proceed at turtle speed. Optimizing and automating software will greatly improve that speed. But as soon as the first photonic processors and RAMs are produced, production should take off like a jackrabbit.

#### DISTRIBUTED DEVELOPMENT

If you had patented an important invention, how would you develop it! Find the most talented people?

Organize them into a productive cooperating team? Locate the prime production facilities? Determine what is the most effective method for beating the competition? Find the best salespersons and the best advertising method?

Where are the most talented people? How do you find them for a project like this, a process that has never been done before, to produce products that don't even exist yet and when the talent is spread all over the earth? The answer lies in the very process by which photonic computers are made and the lessons of recent history.

Synthetic holograms are essentially made out of data. These days, data can be transmitted virtually anywhere. Such data is produced by programs, written by programmers who commonly work alone or in very small groups. Therefore, there is no reason whatsoever that they need to be located at a central place. Everything can be coordinated via telephone, modem, and fax. Thus, the develop-

**At only \$21.95\*  
for twelve issues...**

**...the Computer  
Applications Journal  
could be the best  
investment you make this year.**

\*Domestic price. Canada @ Mexico add \$10, all other foreign countries add \$28. U.S. funds drawn on U.S banks only.

With your subscription to the *Computer Applications Journal* you will receive twelve great issues plus **valuable discounts** on the *Circuit Cellar Project File* book series. With each issue you can count on:

- ▶ the wit and wisdom of **Steve Ciarcia**
  - ▶ high-level tutorials
  - ▶ hardware design and selection tips
  - ▶ clearly explained firmware design
  - ▶ documented application software
- and more!*

Don't miss a single issue! Mail in the subscription card on page 17 of this issue or **FAST FAX** it to:

**The Computer Applications Journal**  
attn: **Subscriber Services**  
**Fax: (203) 872-2204**

**ACCLAIMED AS  
THE TECHNOLOGY  
RESOURCE  
OF THE '90s!**

ment of holographic computers lends itself very well to a distributed development process.

The personal computer revolution was not the creation of one big company, even though there were some big players. It came about by the hand of thousands of innovative entrepreneurs—little people with big ideas. Together, these people created a revolution that has changed the world and seen the lowly PC capture the vast majority of the computing market. It worked then, and will work now in the transition from electrons to photons.

And what about advertising and sales? There is no better advertising than to offer products that outperform the competition. The excitement that creates is the best advertising.

How do we enlist the help of so many talented people? We pay them! Not by the hour, but by the product sold. As new products are developed, we will evaluate them in a coordinated effort to find out who is able to produce the best products for including in our sales network (including software products.) When accepted, arrangements will be made for manufacturing, distribution and sales. Just as with any other business, each one does his part, and each one shares in the fun and profit.

## PROTECTING YOUR INVESTMENT

Programmable lightware, when used with a laser, functions as active patented photonic transistors. As such, the computer-generated graphics you develop may have important economic value. Improper use or disclosure of them may seriously hinder your ability to secure intellectual property rights to those images and the use thereof.

By law, no one has the right to make, sell, or use a photonic transistor, or the patented process that makes them work, without a license from the Rocky Mountain Research Center. Thus your output images can only be used as actual functioning components (even for private laser testing), if you are a licensed part of our developing and marketing network. That is a protection for you from unlawful competition to any product you work with us to develop.

Listing 1—The following short programs are simple examples of how photonic transistors are implemented.

### Constants:

Al = antialiasing factor = 2.5  
 Lp = lines/line-pair = 2  
 Mm = mm/meter = 1000  
 Sm = Synthetic wavelength multiplier = 16  
 w = wavelength = 632.8 \* 10<sup>-9</sup> meters for a HeNe laser  
 PX = pixels on any 1 side of square = 319 (0=1st pixel position)  
 Sx = pixels/full horizontal screen width = 640  
 Ss = full horizontal screen width = 254 mm for a 10" wide screen

### Useful formulae:

Pm = pixels/meter = (Mm \* Sx/Ss) = 2520 (rounded to nearest pixel)  
 Sw = synthetic wavelength = w \* Sm  
 Ur = useful resolution = (Sx \* Mm)/(Al \* Lp \* Ss) = useful lines/m  
 Da = max deflection angle = arc sin(Sw \* Ur)

### Combining these:

Da = arcsin((Sx \* Mm \* w \* Sm)/(Al \* Lp \* Ss))  
 so:  
 Da = arcsin((640\*1000\*632.8\*10<sup>-9</sup>\*16)/(2.5\*2\*254)) = 0.29°

f = focal length in pixels = (PX \* tan(Da)) / 2  
 Z = synthetic distance between input squares and output square in pixels = 2 \* f = PX \* tan(Da) = 63,222 pixels

D = distance from input pixel (X,Y) to output pixel (I,J)  
 = SQR((X-I)<sup>2</sup> + (Y-J)<sup>2</sup> + Z<sup>2</sup>)

Ph = Phase = sin(2πD(Sw/Pm))  
 IA = Instantaneous amplitude = Ph \* amplitude from input pixel  
 OA = Output amplitude at each pixel = algebraic sum of all instantaneous amplitudes from all input pixels = IA1 + IA2 + IA3 . . . IAn  
 PI = 3.14159 . . . . .

```

PROG 1: for plain laser input, initialize input pixels to
        1...otherwise load an input table from a previous run.
FOR X = 0 TO PX
FOR Y = 0 TO PX
INPUT_TABLE(X,Y)=1
NEXT Y,X
  
```

```

PROG 2: to initialize an output table
FOR I = PX/2 TO PX+PX/2 'because the output square is offset by
        half a square
FOR J = 0 TO PX
OUTPUT_TABLE(I,J)=0
NEXT J,I
  
```

```

PROG 3: to calculate output table from input table A
FOR I = PX/2 TO PX+PX/2
FOR J = 0 TO PX
IA = 0
FOR X = 0 TO PX
'to calculate from the B input square, use FOR X = PX TO 2 * PX
FOR Y = 0 TO PX
DISTANCE = SQR((X-I)2 + (Y-J)2 + Z2) '3D Pythagoras
IA = IA + SIN(DISTANCE * 2x / S) 'instantaneous amplitude
NEXT Y,X
OUTPUT_TABLE(I,J)=IA * INPUT_TABLE(X,Y)
NEXT J,I
  
```

```

PROG 4: to combine the inputs from square A and square B to
        determine the output result when both beams are on.
FOR I = PX/2 TO PX+PX/2
FOR J = 0 TO PX
COMBINED_OUTPUT_TABLE(I,J)=OUTPUT_TABLE_A(I,J) +
  
```

(continued)

Listing I-continued

```
OUTPUT_TABLE_B(I,J)
NEXT J,I
```

PROG 5: to make an input hologram by making selected pixels black.

```
FOR X = 0 TO PX
FOR Y = 0 TO PX
IF OUTPUT_TABLE(X+PX/2,Y) < 0 THEN INPUT_TABLE(X,Y) = 0
NEXT Y,X
```

PROG 6: to make an output DI mask by making selected pixels black.

```
FOR I = PX/2 TO PX+PX/2
FOR J = 0 TO PX
IF COMBINED_OUTPUT_TABLE(I,J) <> 0 THEN MASK_TABLE(I,J)=0 ELSE
MASK_TABLE(I,J)=1
NEXT J,I
```

PROG 7: to make an output CI mask by making selected pixels black.  
V = a cut off value to be determined by needs of device.

```
FOR I = PX/2 TO PX+PX/2
FOR J = 0 TO PX
IF ABS(OUTPUT_TABLE(I,J)) < V THEN MASK_TABLE(I,J)=0 ELSE
MASK_TABLE(I,J) = 1 'if there's light there, let it through.
'notice that both positive and negative phases are there, and that
may, or may not be good, depending upon the design of the follow-
ing device.
NEXT J,I
```

PROG 8: to make a mask with predetermined properties

```
FOR I = PX/2 TO PX+PX/2
FOR J = 0 TO PX
'select the criteria for each pixel as to whether it should be
masked or not based on how that particular pixel behaves given the
three lighted input states of: beam A on, beam B on, and both
beams on.
IF OUTPUT_TABLE_A(I,J) = (beam A on criteria)
IF OUTPUT_TABLE_B(I,J) = (beam B on criteria)
IF COMBINED_OUTPUT_TABLE(I,J) = (both input beams on criteria)
...MASK_TABLE(I,J)=0 ELSE MASK_TABLE(I,J)=1 'depending upon above
criteria
NEXT J,I
```

PROG 9: to superimpose a mask onto an output table

```
FOR I = PX/2 TO PX+PX/2
FOR J = 0 TO PX
IF MASK_TABLE(I,J) = 0 THEN OUTPUT_TABLE(I,J)=0 NEXT J,I
NEXT J,I
```

Note that the output through each mask must be checked in each of the possible input states. Masked should be designed to produce only a single output image from the three input images.

PROG 10: to display the results (at position A)

```
FOR I = 0 TO PX
FOR J = 0 TO PX
IF TABLE_TO_BE_DISPLAYED(I,J) <> 0 THEN PSET(X,Y) 'turn on the
pixel
'PSET (X+PX,Y) for position B
NEXT J,I
```

PROG 11: to make the output of one transistor equal one of the inputs of another transistor

```
FOR X = 0 TO PX '(for input B use PX to 2*PX)
I=X+PX/2 '(for input B use I=X-PX/2)
FOR Y = 0 TO PX
INPUT_TABLE(X,Y)=OUTPUT_TABLE(I,Y)
NEXT Y,X
```

The actual patented process involves the use of any means for producing interference fringes, and then using any means for separating those fringes into their component parts. That coverage is very broad. However, computer graphics (synthetic holograms and masks) made and tested by simulation, as described above, are not covered by patent law because they are just simulations.

By becoming a part of our network, your product will be both protected and defended. What's more, those who are a part of the coordinated network will have access to the latest information so that they will always be on the leading edge, not duplicating the work of others.

So whether you prefer to write the synthetic-hologram-producing software, or produce holograms using this method, you may wish to have your work evaluated as a potential product for sale through our network. Neat idea, huh? You can find out more by accessing our new Photonic Transistor BBS at (406) 273-4692. Call in often, as new material will be continuously added as photonic development proceeds. Our BBS supports data rates up to 14.4 kbps; and uses 8,N,1. □

*John Hait is an electronics engineer with over 35 years of experience in basic circuit design and troubleshooting electronic and data processing systems. He is best known by his books and articles on practical applications of thermodynamics and electrochemistry. He is the inventor of the photonic transistor, which is only one of 85 inventions that span the wide spectrum of practical physics.*

## CONTACT

Rocky Mountain Research Center  
P.O. Box 4694  
Missoula, MT 59806  
(free catalog and network  
information available)

## I R S

401 Very Useful  
402 Moderately Useful  
403 Not Useful

# In the Eye of the Beholder

## Displaying Messages with the LightStick II

If you've ever seen multiple images of the current time as you sweep your eye past a digital clock, you've experienced the premise behind the LightStick II. Find out how to display your own ghostly images.

### FEATURE ARTICLE

Shaun J. Greaney



You walk into the room and, tucked away in a corner, you spy a vertical column of LEDs flickering just slightly. "What's that thing supposed to do!" you ask yourself. Shrugging, you redirect your gaze to more interesting matters. As you turn away, you think you see something from the corner of your eye, but then it's gone. You look back at the LEDs to find them still flickering ever so innocently. Turning away, you again catch a glimpse of a picture. You've just discovered the LightStick.

Bill Bell's United States patent [1] calls his light stick a "Momentary Visual Image Apparatus." It is a device that creates an optical illusion that is formed by multiplexing the multiple columns of an image onto a single column of LEDs. Everyone has seen examples of  $m \times n$ -pixel arrays used to create two-dimensional images. PC graphics adapters support 1024 x 768, 800 x 600, 640 x 480 to name a few. Old dot matrix printers supported 5 x 7 characters. From the instant replay screen in the football stadium to the graphics display on my HP 48SX, they all share something in common. The images all consist of static arrays of  $m$  columns of  $n$  picture elements. The specific spatial displacement in  $x$  and  $y$  required by each individual pixel is provided by the physical display.

What a light stick does differently is to display all  $m$  columns of the picture on a single column of  $n$  pixels.

Construction of the image depends on the human eye's persistence of the retina and on its natural, rapid movements from one point of fixation to another.

The persistence of the retina gives you after-images from flash bulbs and allows you to draw line images with Fourth of July sparklers. The rapid eye movements are involuntary and you are probably unaware of the fact that you are doing it. Normal visual activity consists of fixating on a point of interest, capturing the visual information, and then rapidly moving to the next point of interest. During the transition from one point to the next, normally no visual information is gathered. Everything would appear a blur and is ignored. It is during this time that the light stick works.

If you look directly at the column of LEDs, you will perceive it as a flickering or shimmering stick of light. As the individual pixels are rapidly turned on and off it appears that they are either continuously on, or off, or being slightly modulated in intensity. If instead you ignore the stick and let your eyes wander around the room, the horizontal motion of your eye will spread the columns across your retina, thus building up an image that exists as long as your persistence of the retina holds it.

My light stick was inspired by one of Bill Bell's light sticks that was on display at the Exploratorium in San Francisco back in 1986. The Exploratorium is a nonprofit science museum that everyone should visit at least once, and Bill Bell's light stick optical displays are certainly among the most striking and interesting pieces of modern art.

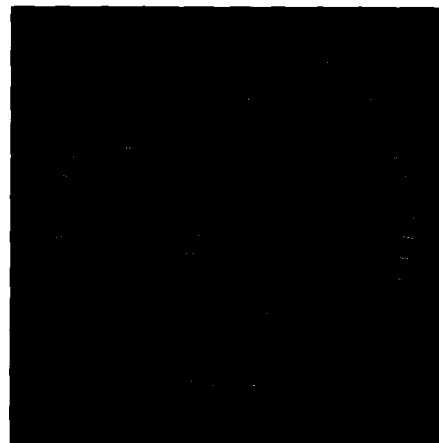


Photo 1—The image produced by the LightStick II is painted on your retina as you quickly look past the device. While it's easy to see the image here, it's gone in an instant when you look at the real display.





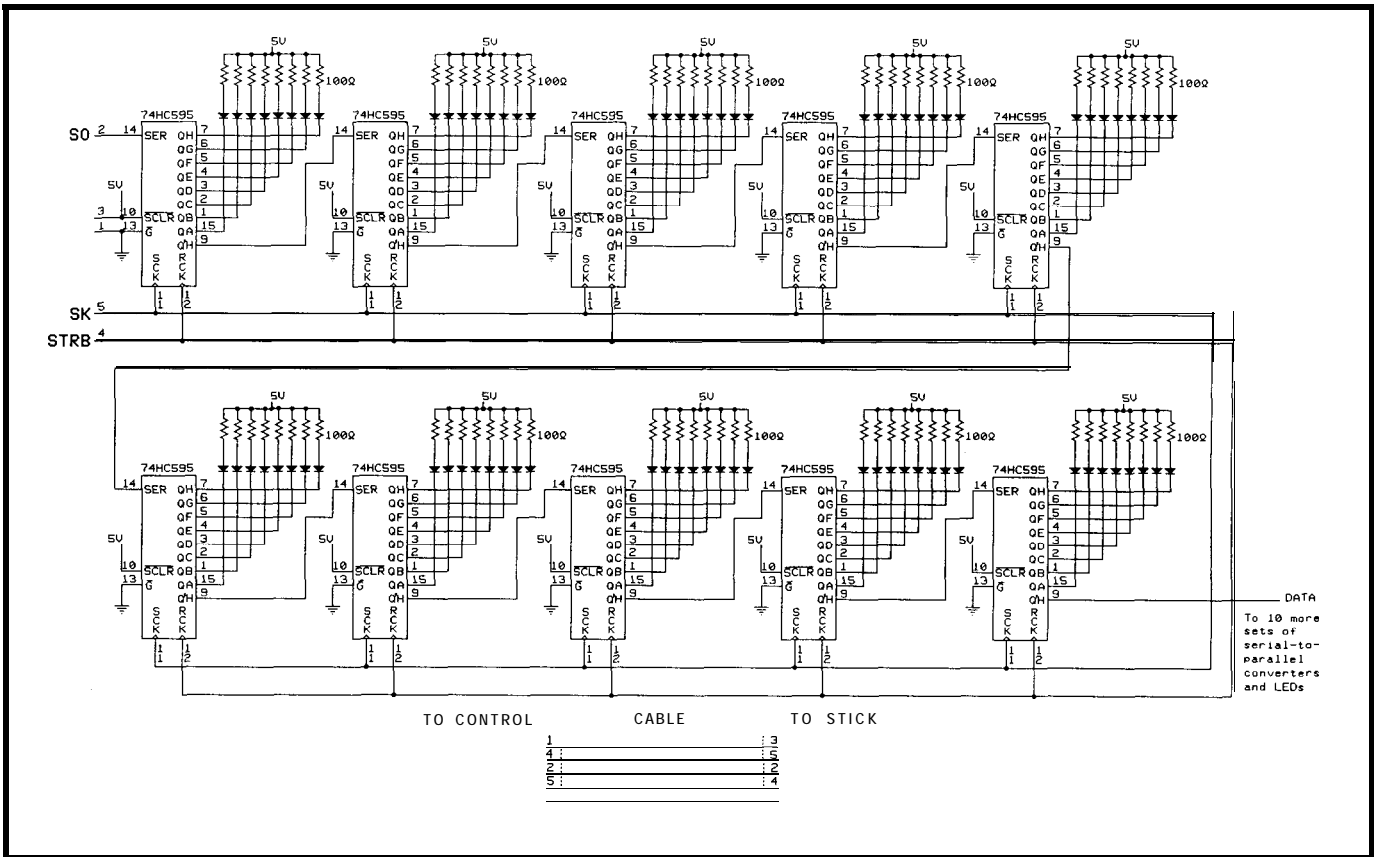
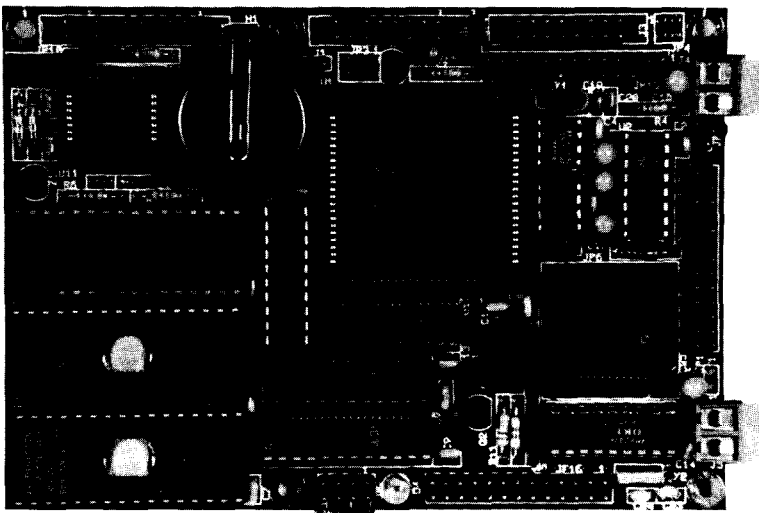


Figure 1'b--Twenty serial-to-parallel converters (only half are shown here) are used to receive data sent by the controller for display on the LEDs.

## V25 POWER COMES TO EMBEDDED CONTROL!

Micromint's RTCV25 is the perfect marriage of an 8088-compatible processor, programming convenience, and control I/O. Forget the need for cross-assemblers and cross-compilers; your favorite high-level language for the PC and a "locate" facility are all you need. The RTCV25 enhances the V25's power with parallel I/O; A/D conversion; RS-232 and RS-485 serial ports; up to 384K of RAM and EPROM; a battery-backed clock/calendar; 128 byte EEPROM, ROM monitor, and the RTC stacking bus. Ease of code development combined with its small size and low power consumption make the RTCV25 ideal for all embedded control applications. And of course the RTCV25 is compatible with Micromint's full line of RTC peripheral boards and products.



### Features:

- 8 MHz CMOS NEC V25 processor
- Up to 384K of RAM and EPROM
- Two serial ports
- Battery-backed clock/calendar
- 128 byte EEPROM
- 40 parallel I/O lines
- d-channel, 8- (or 10-) bit ADC
- RTC stacking bus
- Small 3.5" x 5" size
- 5-volt only operation (150mA max.)
- Full featured ROM monitor
- ROMable operating system

100 Qty.  
OEM configuration

**\$270**



**MICROMINT, INC.**

4 Park St. • Vernon, CT 06066

call 1-800-635-3355

(203) 871-6170 • Fax: (203) 872-2204

ured as Serial Clock, Serial Output, and Serial Input. Since I am not interested in receiving serial data, I grounded G6, the Serial Input. The lower 4 bits of the G port are used to control the addressing of the EPROM (with help from the 74HC4040) and to control the strobe signal to the 74HC595s. The strobe signal is also used to activate the 5-key keypad. I use the L port for data input from both the 8-bit-wide data from the EPROM and the 5-bit-wide keypad.

The image size of 204 x 160 (4080 bytes) allows two uncompressed images in the 64K-bit ROM. Keep in mind that the final image exists only in the persistence of the viewer's retina, so you want to keep it as narrow and as simple as possible.

The keypad is used to control the display of the image. I set a default time delay between column strobes that is a compromise between how narrow the image will appear and how much the leading edge of the image fades by the time the trailing edge is displayed. You may adjust this time

delay using the "faster" and "slower" keys on the keypad. The three other keys allow you to invert the image, choose between the two images in the ROM, and toggle between Normal and Slow mode. In Slow mode the controller clicks off a column every 100 ms (adjustable) for diagnostic purposes. Also, in Slow mode, pressing both "faster" and "slower" together puts you into a Halt mode where you can single step through the ROM one column at time.

### THE POWER SUPPLY

The HP HDSP4830 LEDs are rated at a maximum DC forward current of 30 mA. At 20 mA, the typical forward voltage is 1.6 V. I chose to run them at the maximum DC current to get as much light out as possible while still allowing for the Halt mode where the LEDs may be illuminated for an indeterminate period of time—a 100% duty cycle. I make the assumption that the forward voltage will approach 2.0 V at 30 mA. This leads to a current-limiting resistor of 100Ω. Fortunately,

each current-limiting resistor only dissipates a maximum 116 mW, so I chose a resistor array package, making the assembly more elegant. The LEDs are conservatively rated; I plan to use them in pulsed mode most of the time, and, most importantly, I socketed them.

Assuming a worst-case LED current of 34 mA, 160 LEDs on would draw 5.44 A! This is not counting the shift registers and control circuits. I chose a supply that sources 6 A at 5 V.

### SOFTWARE

The COP800 cross-assembler listing for the most recent version of microprocessor code is available on the Circuit Cellar BBS. The comments in the beginning include the values stored in memory locations unique to the COP8722 EEPROM emulator. Address 804 prevents modifying the ROM section once programmed. Address 805 contains configuration information relating to the mask-selectable oscillator options: divide by 10 or 20 and either RC, crystal, or

## EXPRESS CIRCUITS

MANUFACTURERS OF PROTOTYPE PRINTED CIRCUITS FROM YOUR CAD DESIGNS

TURN AROUND TIMES AVAILABLE FROM 24 HRS — 2 WEEKS

### Special Support For:

- TANGO.PCB
- TANGO SERIES II
- TANGO PLUS
- PROTEL AUTOTRAX
- PROTEL EASYTRAX
- smARTWORK
- HiWIRE-Plus
- HiWIRE II
- EE DESIGNER I
- EE DESIGNER III
- ALL GERBER FORMATS

- FULL TIME MODEM
- GERBER PHOTO PLOTTING

WE CAN NOW WORK FROM  
YOUR EXISTING ARTWORK BY  
SCANNING. CALL FOR  
DETAILS!

*Express*  
Circuits

1150 Foster Street • PO. Box 58  
Industrial Park Road  
Wilkesboro, NC 28697

Quotes:  
1-800-426-5396  
Phone: (919) 667-2100  
Fax: (919) 667-0487

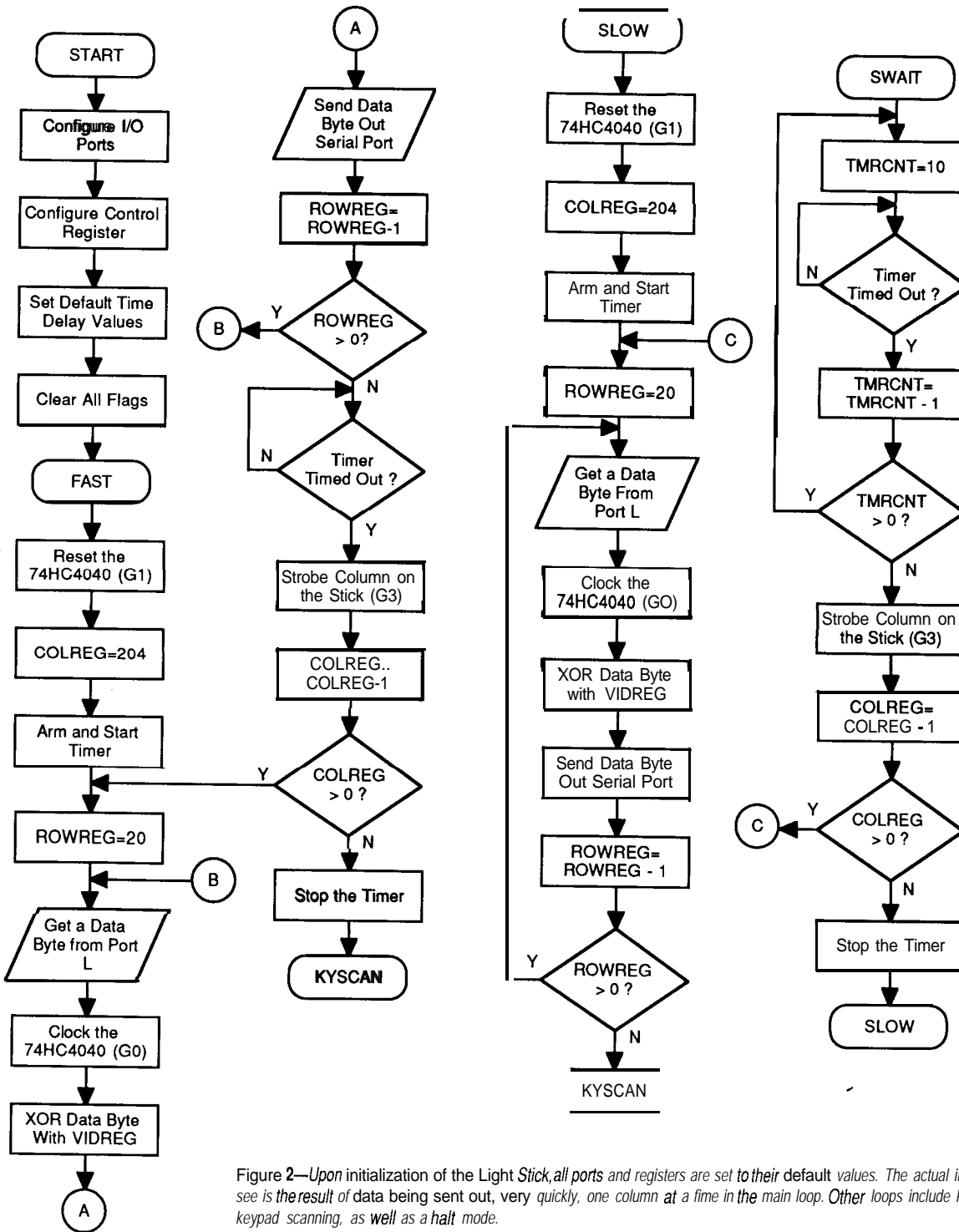


Figure 2—Upon initialization of the Light Stick, all ports and registers are set to their default values. The actual image you see is the result of data being sent out, very quickly, one column at a time in the main loop. Other loops include key and keypad scanning, as well as a halt mode.

external clock. I chose crystal and divide by 20 to give me a 1- $\mu$ s clock with a 20-MHz crystal (the fastest instruction clock possible). Locations 800 through 803 hex are available for comments. I used them for the date and iteration number of the code.

The code may be logically broken up into five routines: an initialization

routine where default values are set; two display loops, one fast and one slow; a keypad servicing routine; and an auxiliary halt mode loop. See Figure 2 for a flowchart of the whole system.

### INITIALIZATION

This routine is run upon power up, hard reset, or in the event of a

software interrupt. I have no provisions for using software interrupts, so if one occurs, it is probably a good idea to reinitialize everything. This routine configures ports L and G, configures the Control Register, and sets the following conditions: do not invert the outgoing data, there is no current keypress, the column time delay for

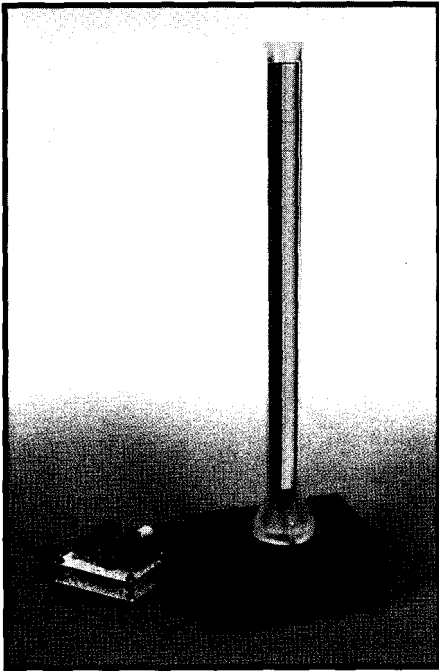


Photo 2—The complete LightStick II consists of a controller board and the display connected by a 5-wire cable. The display stick stands a little under two feet tall.

Slow mode is 100 ms, the column time delay for Fast mode is 466 us, and the serviced flag is set.

### MAIN LOOP

The main loop's primary responsibility is to extract one column of picture data from the 27C64 and shift it into the 74HC595s as quickly as possible so as to allow for maximum variability in column timing. After shifting in an entire column, it waits for the column timer to expire before strobing the new column onto the stick and starting over again.

Optimizing the routine for speed means optimizing for the minimum number of clock cycles used by the instructions within the Fast loop. For this reason, keypad scans are made outside the loop, in between picture displays. Register Indirect instructions only take one clock cycle as opposed to three clock cycles for Memory Direct instructions. If two or more instructions are going to operate on the same location in memory (Port G, for instance), it makes sense to spend the three clock cycles it takes to load the address of the memory location into register B, thereby minimizing the overall total number of clock cycles.

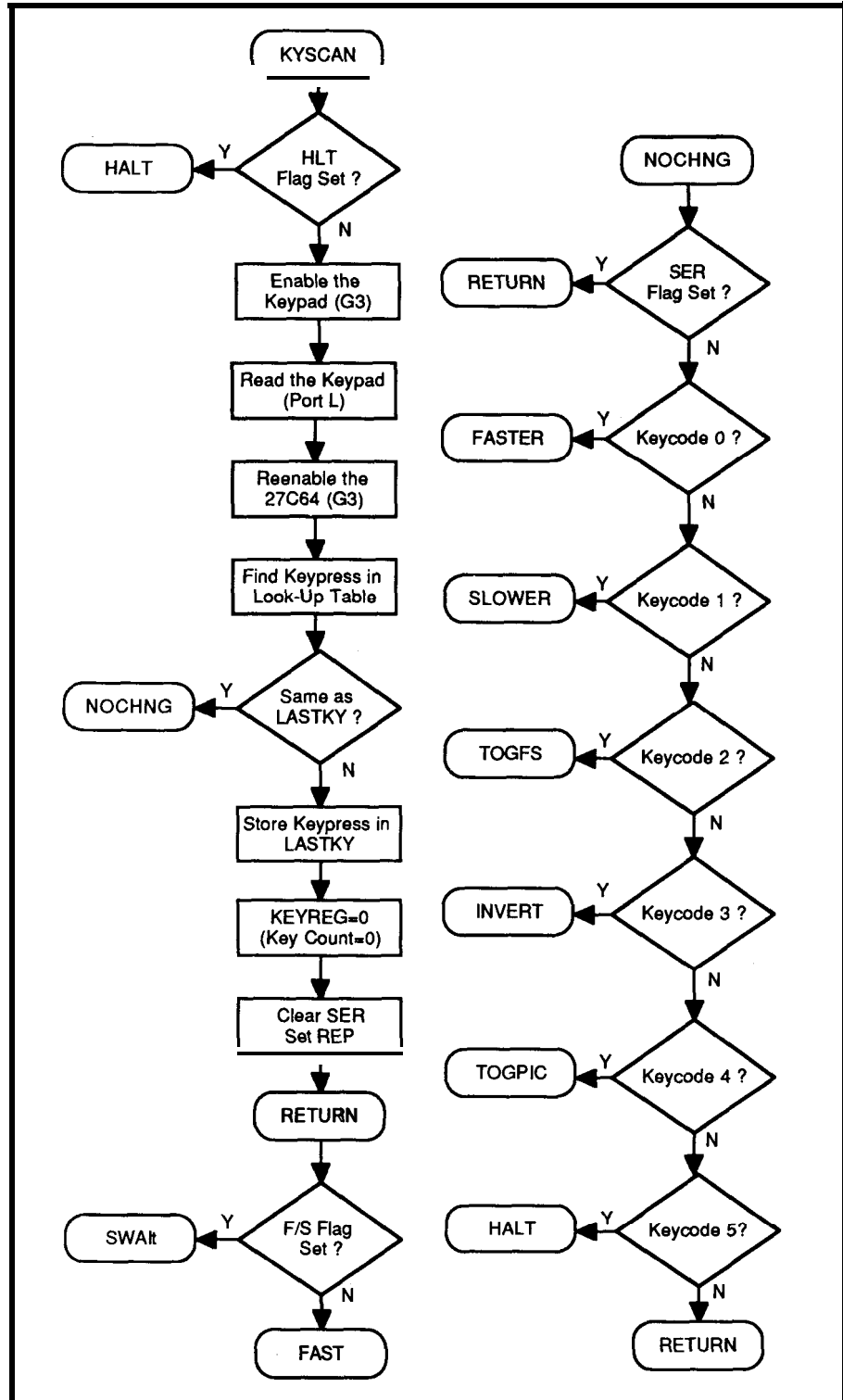


Figure 2—continued

I/O savings are realized by using the 74HC4040 ripple counter to sequentially step through one half of the 27C64's address space. Since both the ripple counter and the 27C64 take a little while to stabilize, I clock the 74HC4040 immediately after getting a data byte to maximize the time available for this settling.

### SECONDARY LOOP

The secondary loop performs basically the same as the main loop except it does a key scan after every column instead of after each complete picture since I am not as time constrained.

This loop is really slow, so it counts up 10 timer periods instead of a

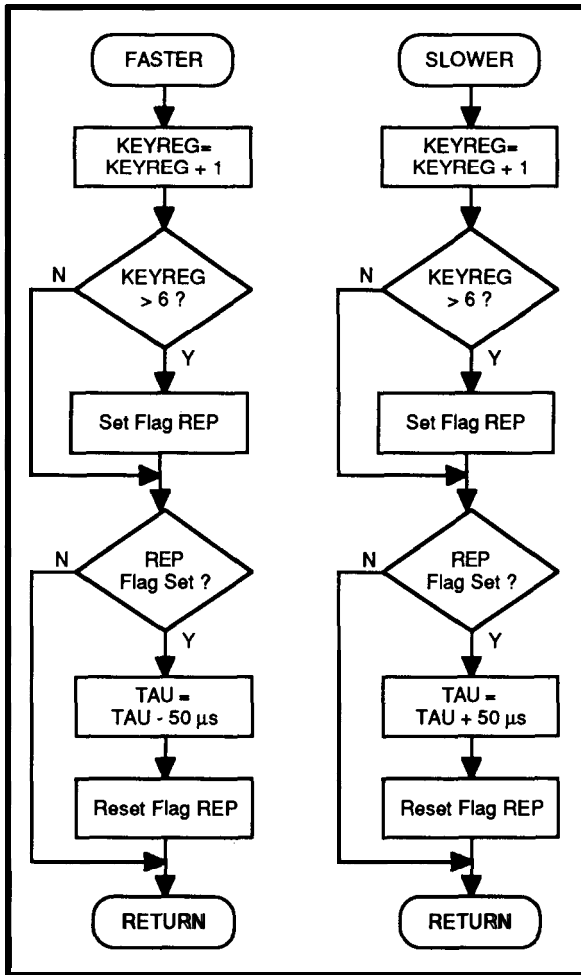


Figure 2-continued

single timer period for the main loop. This also changes the effective timer increment/decrement time slice from 50 μs to 500 μs and allows me to use the same time delay adjustment routine for both the Fast and Slow time delays.

## KEYPADSCAN

The primary responsibility of the Keypad Scan routine is to get updated configuration information in between pictures in the Main Loop or in between columns in the Secondary Loop. The types of things you want to adjust are how quickly you walk through the columns, which picture you want to display, whether or not to invert the picture ("on" for "off" and "off" for "on"), and which primary mode you want the processor to operate in: Fast, Slow, or Halt.

To help me keep track of where I am in the Keypad Scan, I defined eight flags. **REP** helps me keep track of a key that is continuously held down

indicating I should repeat the modification to the time delay. **SE R** keeps track of whether the key request has been serviced. **H LT** indicates Halt mode, used for branching back into the Halt routine. **PI C** indicates which picture is current. **I NV** keeps track of whether or not to invert the outgoing data. **F / S** indicates either Fast or Slow mode, used for branching at the end of the key scan routine. **S LW** is a Slower request, so I have to increase the time delay. **FST** is a Faster request, so I have to decrease the time delay.

## HALT MODE

Halt mode may almost be considered a limited functionality subroutine of the keypad routine. When in Halt, the keypad is continuously examined for one of only two valid keypresses: Step or Continue. Just as the regular Keypad routine either jumps back to Fast or Slow depending on where you came from, when you get a Step keypress you go back to the parent process and execute only one column for Slow or

one picture for Fast. When you go for another key scan you are placed back in Halt. The only ways out of Halt are Continue and Reset.

## TRANSLATION UTILITIES

Since the column-at-a-time, 204 x 160, two-color picture format is not an accepted standard yet, a couple of C-code programs were written to translate data from the existing, more common standards into STK-light stick-format. Both of these programs are available on the Circuit Cellar BBS.

The company logo image data was given to me in an unformatted 336 wide by 150 high two-color binary file that was organized as a sequence of rows starting from the upper left corner of the picture. That's 6300 bytes of input picture data. I had to convert the data into a stream of ASCII represented hex bytes that were grouped one column at a time starting from the upper left hand corner of the picture.

A friend at work, Tom Recht, wrote the original translation program that took the raw binary input file and rearranged it for LightStick I (yes, there was a version before this one). I modified this program to create **X LATE 2 . C**, the LightStick II translation program.

Since the stick is 160 rows high, a 336 x 160 or 6720-pixel buffer is created and initialized. The binary input file is then read into this 6720

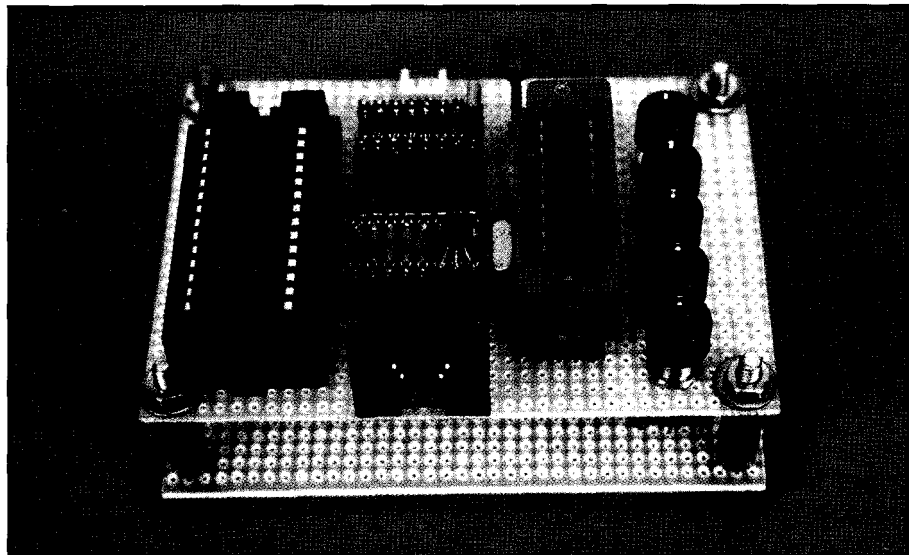


Photo 3—The controller board has ZIF sockets for the the processor and EPROM to make it easier to change the control code and the image displayed.

array of bytes where each byte represents a single pixel of the picture. The picture is stored in the original data file such that the most-significant bit of the first byte is the upper left corner of the picture, and the buffer is arranged such that the first byte is the upper left corner so that the index loops used to write the data can start at zero and increase monotonically to read from left to right and top to bottom.

The company logo is not exactly symmetrical, so the first half is only 160 columns wide (20x8) with real data, and the file is padded with 896 null bytes to pad the 27C64 EPROM to 4096 bytes for the first half of the logo. The second half has 176 columns of real data (22x8), so it only needs 576 null bytes of padding to fill the second 4096 bytes of the EPROM.

The output file is written as ASCII represented hex bytes. Another friend at work, Bruce Haggerman, wrote a C utility that takes such a hex file and adds the appropriate addresses and checksums to create an Intel hex format file ready to download to the PROM programmer.

Eventually, the company logo gets kind of old. To give myself some practice coding C, I wrote a program that takes uncompressed, B&W, windows BMP image files, pads or truncates the image to the 204 by 160 light stick size, and generates an ASCII represented hex output file. I found the windows BMP file and image headers explained in "Graphics File Formats." [2]

In BMP2STK. C, I tried to write code that I could easily modify for other programs. To that end there is a general-purpose, configurable file open() routine that tries to get all the input and output file information from the command line arguments. If this fails; it will prompt the user for the information.

The routine of interest from the light stick perspective is read (). This routine defines two structures to extract size and configuration information from the file and image headers. Using information in the headers, I check for valid BMP file type, check for the two-color requirement, and get

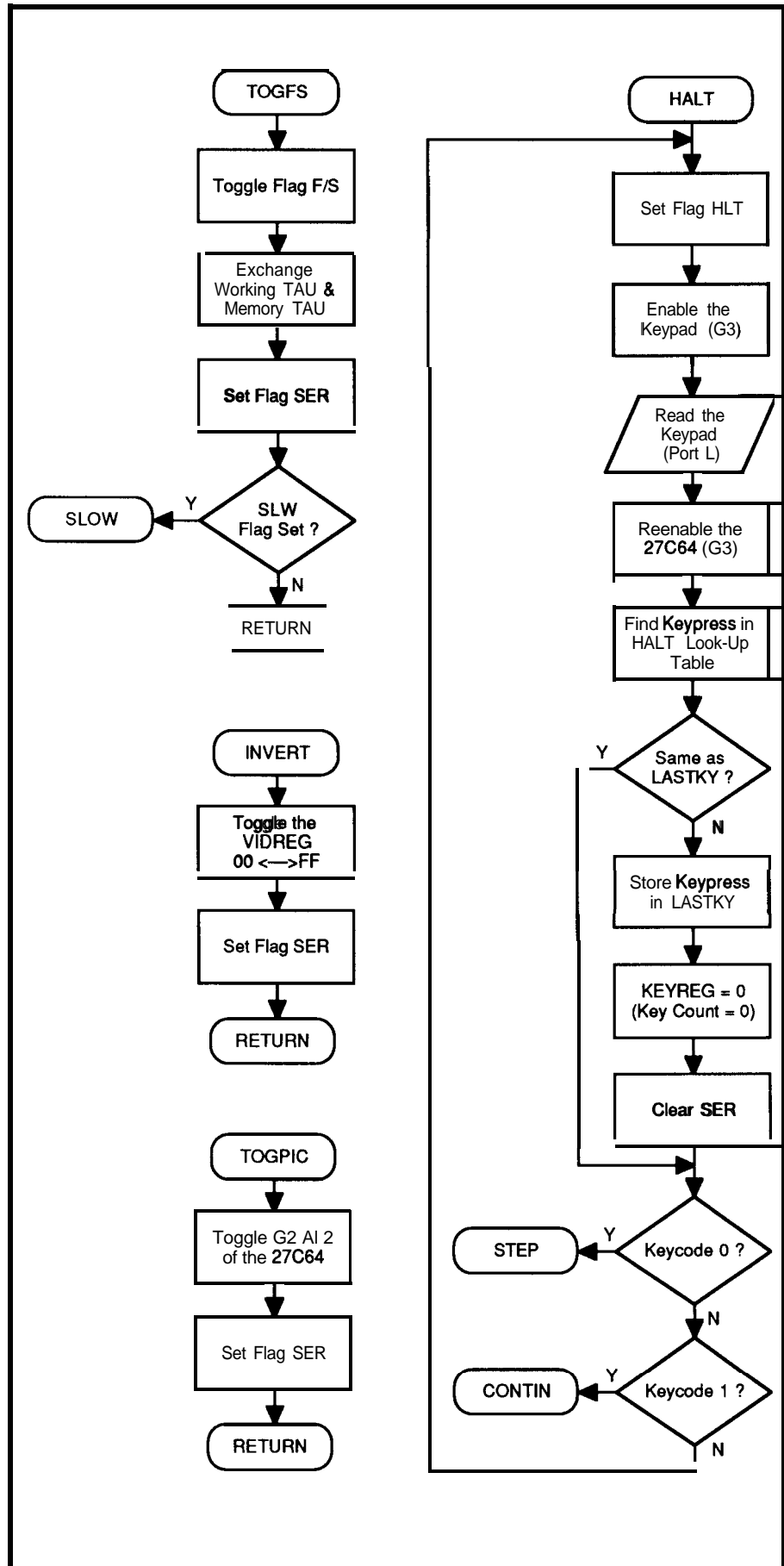


Figure 2-continued

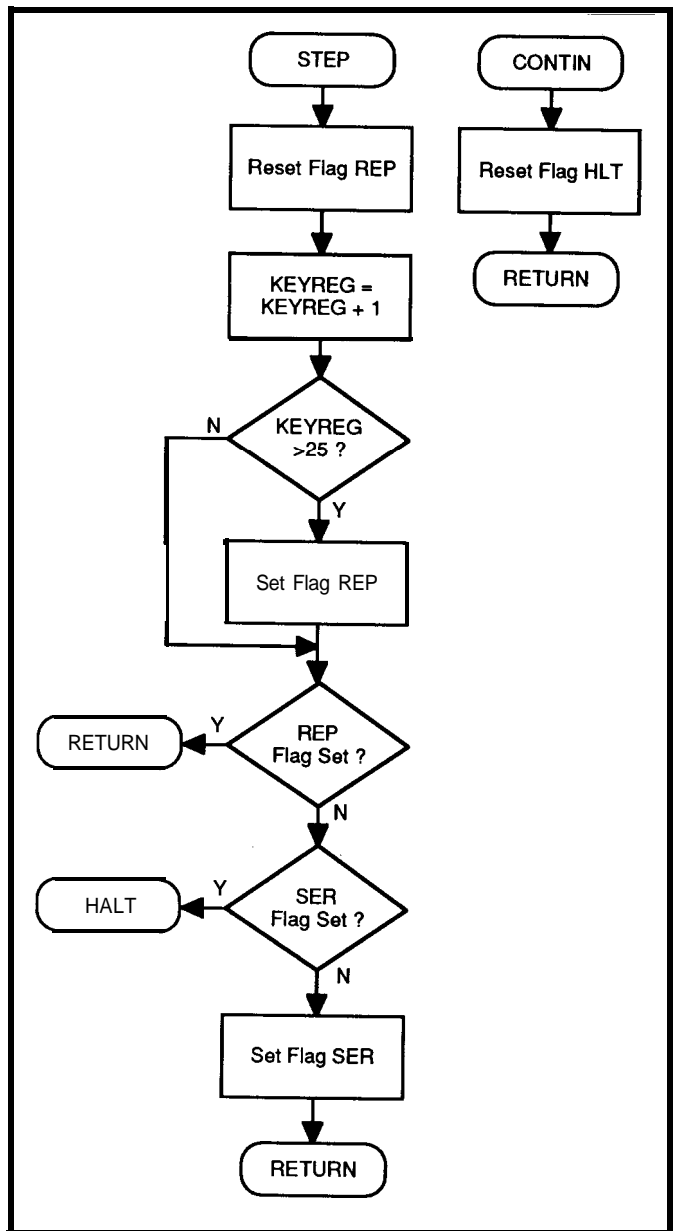


Figure 2—continued

loop sizes from the image size data fields.

The BMP image format requires the data stored in the BMP file to fall on 4-byte boundaries for image width. If the actual image size does not fall onto a 4-byte boundary, the image data is padded with garbage out to the next 4-byte boundary. It is the responsibility of the display program to get the actual image size from the image header and to crop this garbage data out of the displayed image.

I have a 204 wide by 160 high buffer that is initialized to 00. As I read the BMP into the buffer, a row at a time, I check to see if I exceed the buffer width. If so, I keep reading bytes

in and discard them. I store the data into the buffer one pixel at a time, from left to right. If the image is less than 204 pixels wide and not an integer number of 4-byte blocks wide, I stop transferring pixels at the last data bit of significance. Since the buffer is initially set to all 00, the image is effectively padded out to 204 pixels wide. The other interesting thing about BMP files is that they are arranged with the lower left corner of the picture first. This means the upper and right edge of the picture will be clipped if it is larger than 204 by 160.

The write() routine is very similar to that of the XLATE2.C program. It also pads 16 bytes to make up the difference between the 4080 bytes in one 204 x 160 light stick image and the 4096-byte capacity of half of the 27C64 EPROM. This program takes two BMP files as input and generates one STK file as output.

## CONCLUSION

LightStick II provides a fairly flexible architecture for examining the optical effect described by Bell in his patent. This allows you to play with both the type of picture and the display parameters of the picture. Keep in mind that the COP assembly code and the C code were written by an RF engineer, so if you believe there is a more efficient or elegant way of

writing the code, you are probably right. If you do rewrite any of the code to be more flexible, efficient, or elegant, I would be interested in hearing about your efforts.

The type of information that may be successfully displayed is somewhat limited by the humans that are going to observe it. When I first wrote the BMP2STK.C translation code, I feverishly converted a couple of B&W digitized pictures to the stick format. I then spent half an hour getting a sore neck trying to view them. The best images are simple geometric shapes, words with few letters, and stick figures. Even the Circuit Cellar INK logo used in the photo for this article presents a challenging image to view in its entirety. So, don't worry if you are not the most elaborate Paintbrush artist in the world; the people who would be required to view your creation don't exist yet anyway. □

**Shaun Greaney is an RF Design Engineer at AT&T. He holds a BSEE from Rensselaer Polytechnic Institute and an MSEE from the University of California, Berkeley. He may be reached at [sgj@hocpa.att.com](mailto:sgj@hocpa.att.com).**

## REFERENCES

1. Bell, Bill, United States Patent Number 4,470,044, "Momentary Visual Image Apparatus," 1984.
2. Kay, David C. and John R. Levine, "Graphics File Formats," Windcrest/McGraw-Hill, 1992, ISBN 0-8306-3059-7.

## SOFTWARE

Software for this article is available from the Circuit Cellar BBS and on Software On Disk for this issue. Please see the end of "ConnecTime" in this issue for downloading and ordering information.

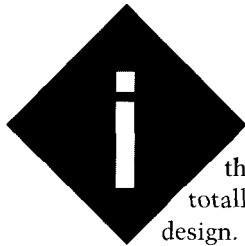
- 404 Very Useful
- 405 Moderately Useful
- 406 Not Useful

# Interrupt-free Design

Over the years, the purpose behind a design decision is often forgotten in a sea of, "But that's the way we've always done it." If you automatically use interrupts in your embedded designs, think again.

## FEATURE ARTICLE

### Do-While Jones



It is getting rare these days to find a totally analog circuit design. Just about anything you build will have some digital logic in it. Unless that logic is so simple that it can be done with a few gates and a flip-flop or two, the logic will naturally be done by a microprocessor. The microprocessor frequently controls peripheral devices that are attached to it with interrupts, and that is often a mistake.

Using interrupts for peripheral control is so universally accepted that it is seldom questioned. Certainly there are times when using interrupts makes good sense, but there are other times when the use of interrupts can make the design more difficult. Knowing how to recognize situations when it is wise to use interrupts or to avoid them like the plague will lead to an interrupt-free design that may be cheaper, simpler, and more reliable.

#### A SHORT HISTORY OF INTERRUPTS

If you know why interrupts were invented, your understanding of what kind of problems they solve will help you decide if they are beneficial to your design or not. So, let's return to the early 1960s and reminisce about what it was like to use an IBM 1620 computer.

In those days, clock speeds were in the tens of kilohertz (not tens of megahertz). The bits on the data bus changed so slowly that the IBM 1620 had a speaker attached to one of the data lines and you could hear the bits flip. After a while you could tell whether your program was operating

correctly by the sound coming from the speaker.

The 1620 had only one terminal, one card reader, and one card punch. It could only run one program at a time. But there were many people who wanted to use the computer and they didn't want to wait in line to submit their decks of punched cards. So, someone got the bright idea to connect multiple terminals to a single computer and make it serve them all simultaneously (or at least, apparently simultaneously).

At first, I'm sure someone must have tried to poll the terminal to see if a user had pressed a key. They possibly tried to poll each terminal ten times a second to see if the user had pressed a key, but in those days it probably took about 20 clock cycles to poll a terminal's status register. Suppose you poll 15 terminals 10 times per second on a computer with a 10-kHz clock. Twenty cycles at 10 kHz is 2 milliseconds. Multiply that by 15 terminals 10 times per second, and you will see that 300 milliseconds of each second would be used just to see if any key has been pressed. Nearly a third of the CPU time was wasted just checking to see if any key had been pressed (and discovering that one had not been pressed). Computer time was too valuable to be wasted that way.

About the same time user terminals were invented, punched cards were being replaced by magnetic files on drums or disks. Bits were read from a sector as the magnetic media passed under the read head. The bits were stored in a buffer and had to be written to the proper memory locations before the next sector could be read. If the buffer wasn't flushed in time, the computer had to wait for a complete revolution before the next sector could be read. The computer couldn't do anything else while the buffer was being filled because it had to be ready to empty the buffer as soon as the buffer was filled. This also wasted valuable computer time.

Interrupts were invented to solve these two problems. They eliminated the need for the computer to waste time checking all the terminal ports because the terminals could interrupt



whatever the processor was doing to get attention. The computer could process data while sectors were being read into the disk (or drum) buffer because the buffer would interrupt the computer when it was full and needed emptying. People learned that if they wanted fast response time and low overhead, interrupts should be used.

Over the years, it became standard practice to use interrupts to interface peripherals to computers. In a general-purpose, multiuser computer, that is still the wisest thing to do. Interrupts (and the associated file control blocks) make it easy to configure operating systems to support a wide range of peripheral devices. They maximize throughput because the computer can do a background job when the peripherals don't require attention. The modern operating systems we have today could not be built without interrupts. Interrupts are an essential part of general-purpose computing.

## DIFFERENT NEEDS

Most of the circuits designed are for special-purpose applications. The microprocessor put in a microwave oven, VCR, or television set does not need a multiuser operating system. It doesn't need a system administrator to add new users or install new peripherals. An embedded system specializes in doing one job and it usually needs to do it virtually instantaneously. It doesn't need the versatility interrupts provide and can't afford to spend the extra time needed to process them.

Some designers don't seem to recognize the difference between a general-purpose computer and a dedicated controller. For example, it is common to see an A/D converter attached to an interrupt handler in a dedicated controller design. This is never a smart thing to do. If you ask the designer why he did it that way he will probably say, "I used an interrupt because I needed fast response time." That's like saying, "I made it out of solid gold because I needed it to be cheap."

An A/D converter satisfies neither of the two criteria for correct use of an interrupt. Interrupts should be used when (1) you can't predict when data

will be available and (2) data must be taken from the peripheral promptly or it will be lost. Dedicated controllers always know when data from an A/D converter will be available. It comes at a periodic rate synchronized to a clock. It should be no surprise that data is available. Furthermore, it is the program that must have the data promptly, not the peripheral that must have the data taken from it promptly.

Consider what happens in a typical dedicated controller if it polls to collect data from an A/D converter. At regular intervals it fetches the data from the last A/D conversion, and processes the

***•J Interrupts should only be used in those situations that are so important, the computer must stop whatever it is doing to respond to them.***

data from the last conversion. When all the processing is finished, it sits in a tight loop "wasting" machine time polling a timer to see if it is time to begin the next processing cycle. [This time isn't really wasted because there is nothing else the controller could be doing.]

If you do the same thing using an interrupt, the processing takes a little longer. At regular intervals the controller fetches data which was stored in memory from the last A/D conversion, starts the next conversion, and processes the data from the last conversion. At some time during this processing sequence the computer is interrupted by the completion of the A/D conversion. The computer stores all the registers on the stack, fetches the data from the A/D converter, and stores it in the same memory location that it just recently fetched data from. Then the interrupt-service routine restores all the registers and returns to the interrupted program. It finishes processing and spends a little less time waiting for the next cycle to start because it had to spend some time

saving and restoring registers. If there was just barely enough time to do the essential processing, the accidental overhead of the interrupt servicing may cause the program to miss its deadline.

The situation is even worse if the dedicated controller uses a processor with a typical operating system. At regular intervals it fetches data which was stored in memory from the last A/D conversion, starts the next conversion, and processes the data from the last conversion. At some time during this processing sequence it is interrupted by the completion of the A/D conversion. The task context is saved and the interrupt-service routine fetches data from the ADC and stores it in memory. Then the operating system spends more time deciding which tasks are eligible to run and picks one of them to run next. If you're lucky, it will decide to resume the task that was interrupted. The context switching time may exceed the essential processing time and there is a very good chance the program will not be able to meet its deadlines.

It is always faster to poll a peripheral's status port to make sure it is ready and then transfer data to (or from) the peripheral's data port than it is to respond to an interrupt generated by the peripheral's status port.

Furthermore, interrupts just annoy the processor by telling it exactly when new data is available. In the example above, the processor doesn't care when the A/D converter finishes its conversion. It only cares that the conversion is complete by the time the program needs that data. There was no reason to force the processor to read the output of the A/D converter and store it in memory the instant the conversion was complete. The A/D converter could have held the data until the processor was ready to use it. Instead, it used an interrupt to usurp control and forced the processor to take the data at an inconvenient time.

What often happens in cases like these is the designer keeps the interrupts disabled almost all the time. The program enables the interrupts only at those times when it needs data. In effect, this is just a less efficient way

of polling the peripheral. (I've even seen designs where the interrupt-enable output was connected to an interrupt input to assure that an interrupt happened every time interrupts were enabled. I am not making this up!)

## PROPER INTERRUPT USE

Interrupts should only be used in those situations that are so important, the computer must stop whatever it is doing to respond to them. For example, the radiation-leak detector in a nuclear power plant should be connected to an interrupt. If radiation is leaking, the computer should stop whatever it is doing and handle the emergency *right now*. The computer should not poll the detector every ten minutes because it can't afford to wait ten minutes to discover the problem. Nor should the computer poll the detector every five microseconds, because then it would not have time to do anything else. Interrupts should be used when a peripheral has something unexpected and vital that must be communicated

to the processor, regardless of what the processor is doing.

If you are still uncertain when it is appropriate to use interrupts, consider this analogy. Suppose you are driving through Arizona on a family vacation. It's OK to poll your children, "Is anybody hungry?" when approaching a restaurant. Depending on their response you can decide if you should make a convenient stop. Don't poll them in the middle of the desert when you are miles from the nearest restaurant. It just encourages the kids to complain, "I'm hungry!" Mild discomfort from hunger isn't an emergency and there's nothing you can do about it anyway, so don't ask. If they do get so hungry that they complain without prompting, there still isn't anything you can do about it, so you have to ignore their annoying interrupts.

It is an entirely different situation when a child in the back seat interrupts you and says, "I think I'm going to throw up!" That is an interrupt that must be handled right away! You must find some way to pull off to the side of

the road as soon as possible. This interrupt should only be masked when you are going down a steep one-lane mountain road with a logging truck right behind you.

## WHO'S THE BOSS?

It comes down to a question of who should be in control. Should the CPU be in control or should the peripherals be in control? There are a few designs where the peripherals should be in control of the program flow and they should be connected to interrupts. A telephone-switching network might be one example where the design should be interrupt-driven because there is no reasonable way to predict how many calls will be placed, when they will be placed, or the order in which they will be placed. The design must adapt to the demand.

Interrupt-driven designs tend to be chaotic and nondeterministic. You have no way of determining when the interrupts will happen, so you have to be very careful to make sure there are no race conditions or sequences of

# µLAN

## The 9-Bit Solution is the Answer to your Embedded Network Needs

PC/XT/AT  
8051  
8096  
80C186EB/EC  
68HC11  
68HC16  
Z180

**The Cimetrics Technology 9-Bit Solution is a complete microcontroller network (µLAN) that supports the 8051, 68HC11, 80186, and many other popular processors. The 9-Bit Solution takes full advantage of multiprocessor modes built into microcontroller serial ports. Our flexible software and hardware allow developers to create powerful, yet inexpensive master/slave multidrop embedded controller networks.**

- Up to 250 nodes
- 16-bit CRC error checking with sequence numbers
- Low network overhead and low resource requirements
- RS-485 interface card for the PC
- Complete source code included
- Comprehensive documentation

**CIMETRICS TECHNOLOGY**

120 West State Street, Ithaca, NY 14850  
TEL: (607) 273-5715 FAX: (607) 273-5712



**ORDERS (800)477-4181** ORDERS(410)437-2324

INFO (410)437-4181 VISA • MASTERCARD • AMEX 1292 MONTCLAIR DRIVE  
FAX (410)437-3757 FEDERAL EXPRESS NEXT DAY AVAILABLE PASADENA, MD 2,122

<b>X-10</b>	HOME AUTOMATION MODULES & CONTROLLERS	
LEVITON	HOME AUTOMATION MODULES & CONTROLLERS	
STANLEY	X-10 COMPATIBLE GARAGE DOOR OPENER	
RCS	X-10 COMPATIBLE 4 CHANNEL DPTP RELAY	\$160
REDAC	ADD DIGITAL INPUTS, ANALOG INPUTS & RELAY OUTPUTS TO X-10	
ECS SOFTWARE	DO IT ALL IN HOME AUTOMATION	\$250
SKYLINE CONTROL	SOFTWARE UPGRADE FOR X-10 CP290P EASIER TO EDIT USE DUSK AND DAWN CONTROL \$49.95 OR \$20 W/CP290P	
APPROACHING HOME AUTOMATION	BOOK ON X-10	\$18.99
UNIVERSAL ELECTRONICS	ONE-FOR-ALL REMOTES	FROM \$11.24
SSI	DOLBY PRO LOGIC SURROUND SOUND DECODERS, AMPS & SPEAKERS	
AR POWERED PARTNERS	COMPUTER SELF POWERED SPEAKERS	
FOSGATE	HTS & IN-WALL SPEAKERS	
PANASONIC	HYBRID PHONE SYSTEM	
PANASONIC	PHONES, FAX & ANSWERING MACHINES	
MAGNAVOX	CLOSED CIRCUIT TV SYSTEM	\$399
RAINDRIP	LAWN GENIE	
	<b>JANUARY SPECIAL</b>	
MOVIE ADAPTER FOR CD1200		\$219
PANASONIC KX-T30810	HYBRID CONTROL UNIT	\$499
PANASONIC KX-T61610	B-LINES 16-EXTENSIONS	\$825
ES1400e	X-10 TWO-WAY COMPUTER INTERFACE	\$349
CD1200	PHILIPS INTERACTIVE CD PLAYER	\$499
ENCYCLOPEDIA INCLUDED \$299 VALUE		MOVIES NOW AVAILABLE
<b>CALL OR WRITE FOR CATALOG</b>		

DEALERS PLEASE WRITE OR FAX ON COMPANY LETTERHEAD

interrupts that can cause trouble. You have to decide when to mask interrupts and try to predict what will happen if some interrupts are masked when an interrupt occurs. This can be done, but it isn't easy to do without making a mistake.

Interrupt-free designs are much more predictable. The program logic controls the data and control flow. The number of possible execution paths is finite and usually small enough to be counted. This makes it possible to determine the worst-case execution time and the maximum stack depth. The logic is simpler because the program controls the order in which things happen. The program runs faster because there isn't any interrupt-service routine overhead.

### THREE EASY RULES

It is worth repeating the two main criteria for deciding when to use interrupts.

- Interrupts should be used when you can't predict when data will be available.

\*Interrupts should be used when data must be taken from the peripheral promptly (because the peripheral can't store it, and data could be lost).

A third, more subjective criterion can even be added: interrupts should be used when they make the design simpler.

If you find you are tearing your hair out trying to decide when to enable interrupts or how to change interrupt priorities dynamically, you probably should not be using interrupts at all. If you were using interrupts appropriately, they would solve your problems, not cause them.

Make each interrupt justify its existence. Ask, "Why can't the CPU just poll this device when the CPU is ready to accept data?" If you can't come up with a good answer to this question, then don't use an interrupt.

### USE INTERRUPT-FREE DESIGN

An interrupt-driven design turns the CPU into a slave trying to serve multiple masters. You have to devise rules that decide what to do when

several peripherals ask for service simultaneously or at an unexpected time. This makes the design more complicated than it needs to be.

You don't need to be a slave to tradition and use interrupts for your peripheral interfaces. An interrupt-free design puts you in control of the logic. Things will happen in the order you specify. An interrupt-free design puts peripherals in their proper place (as servants, not masters). An interrupt-free design makes peripherals hold data until the CPU is ready to do something with it. An interrupt-free design replaces chaos with order. □

*Do-While Jones has been employed in the defense industry since 1971. He has published more than 35 articles in a variety of popular computer magazines. He is the author of one book, ADA in Action.*

## I R S

- 407 Very Useful
- 408 Moderately Useful
- 409 Not Useful

## !NEW! FROM CIRCUIT CELLAR PROJECT FILE, VOL. II

**\$17.95**

**SAVE 28% OFF**

**COVER PRICE OF \$24.95**

**All NEW Projects to SPARK Your Imagination!**

GP28 Audio Sampling System  
Wiring Your House for the 21st Century  
Multiprocessor Architecture using DSP  
**ANDMUCHMORE!!!**

VISA, Master-Card, or International Postal Money  
Order (U.S. funds drawn on U.S. bank only)

## Circuit Cellar Project File

4 Park Street  
Vernon, CT 06066

Tel: (203) 875-2199  
Fax: (203) 872-2204

\*Includes domestic delivery. Please add \$6 per copy for delivery to Canada & Mexico, add \$8 per copy for deliver, to other non-U.S. addresses.

### TIRED OF WAITING FOR THE PROMPT ?

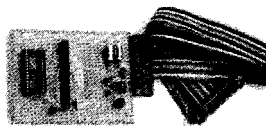
Speed up with a ROM DRIVE! Boots DOS and programs instantly. Also used to replace mechanical drive completely in controllers or diskless workstations. The only perfect protection from viruses. Easy to install half-size card.

MVDISK1 128k \$75  
MVDISK2 720k \$150  
MVDISK3 2.88m \$195

**\$75**

Quantity discounts!

## DOS IN ROM!



## \$95 EPROM PROGRAMMER

- Super Fast Programming
- Easier to use than others
- Does 2764/27080 (8 Meg)

### WORLDS SMALLEST PC !!!

ROBOTS ALARMS RECORDERS DOS

THREE EASY STEPS: **\$27** 1K QTY  
1. Develop on PC  
2. Download to SBC **\$95** SGL QTY  
3. Burn into EPROM

- 2 PARALLEL -LCD INTERFACE
- 3 SERIAL -KEYBOARD INPUT
- PC TYPE BUS -REAL TIME CLK
- BIOS OPTION -BATTERY OR SV

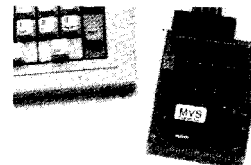
FREE SHIPPING IN U.S.

5 YEAR LIMITED WARRANTY

**.MVS**

Box 850  
Merrimack, NH  
(508) 792 9507

## 8088 SINGLE BOARD COMPUTER

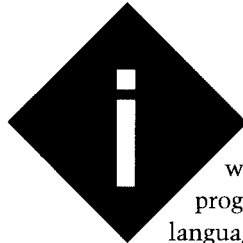


# FEATURE ARTICLE

Hank Wallace

## An Assembly Language Programming Aid

Even though it may seem everyone has switched to high-level languages, experienced programmers know the value of hand-crafted assembly code. Here's one technique that combines the best of both worlds.



I know a guy who writes all his programs in assembly language. On occasion, he has been heard to proclaim, "C! Ack! I can do anything in assembly that you can do in C, and it's faster." Then he'll proceed to flap through his latest ten-thousand liner to prove his point. Most programmers are at the opposite extreme, gritting their teeth at the thought of a large assembly language program. They would rather have a faithful compiler hassle out all the details while they worry about more interesting problems. I don't blame them.

However, for the bold and faint-hearted alike, I will present you with a method that can be used as a tool for writing assembly language programs that can be easily understood and supported. This method deals simply with the nuts and bolts of how we write assembler code; no AI, CASE, 4GL, fuzzy logic, or SDI, just common sense. You won't find it mentioned in the current CASE literature frenzy. It is much more fundamental and practical than that, and has its roots in personal programming discipline. After all, if you give a sloppy, undisciplined programmer great CASE tools, he will still produce sloppy programs, only faster.

What we shall examine is a way to apply the benefits of high-level languages to assembly language programs. The approach is valid since many assembly programmers start with high-level language programming.

When was the last time you worked on someone else's assembly program? Were you able to work comfortably with the code and attending documentation, or did it leave you on the verge of a slanderous rampage? The complaints I have voiced most often (and loudly in an empty lab) are these:

- \*This program does not have enough high-level documentation.
- This program has wasteful documentation of its most simple parts.
- This program is packed with spaghetti code.
- \*There is not enough information here to even assemble the original source (makefiles, etc.), much less make changes.
- \*The comments are written in a grammatically incorrect format, eliminating important words, such as verbs and articles, for the sake of brevity.
- This programmer wasted his time on long-winded comments that are as extensive as the source code. Why shouldn't I just skip the comments and decipher the actual code?
- This flowchart is out of date with reference to the current source.
- \*Add your favorite gripe here.

What is the cure for this disease? More expensive tools? A faster PC? CASE? No, just simple programming discipline, possibly augmented with the Golden Rule: Program unto others as you would have them program unto you.

Let's look at two characteristics of a high-level language which are important in the construction of assembly language programs. The C language is a suitable candidate for our discussion since many use it. First, programs written in C have a fundamental structure. The basic element of the program is the *statement*. State-

ments are gathered into groups called blocks, delimited by curly braces. The execution of each block of statements is controlled by IF-THEN-ELSE, DO-WHILE, WHILE, FOR, and SWITCH constructs, depending on the function of the program. Most procedural languages have many or all of these constructs. The primary point is that they allow the control and collection of statements into groups which themselves may be so grouped, permitting the orderly expression of hierarchical algorithms.

The second important characteristic of C and most high-level languages is that the programmer usually does not have to know anything about the host computer's assembly language to implement an algorithm. This characteristic means the programs are somewhat machine independent. You guessed it—we are going to try to make assembly language programs more portable.

## SWEET AND SIMPLE

The technique described here involves writing an assembly program in two easy steps. First the program is written in a structured, machine-independent high-level language, and then hand compiled into assembly language for the target processor. The text of the high-level language becomes the commentary for the assembly program. Your brain takes the place of the computer-based compiler.

This discipline (oh, that word again) has a number of advantages. The main one is that the algorithm can be conceptually separated from its implementation on a particular microprocessor. In other words, since the algorithm is completed in a high-level language version before assembly coding starts, the programmer design-

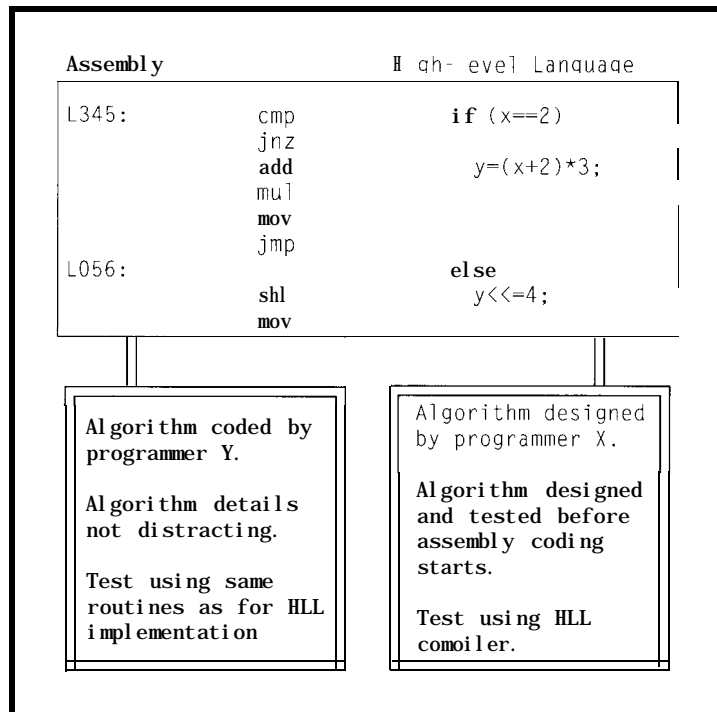


Figure 1—Divide-and-conquer assembly language programming. The high-level/algorithm on the right is separated from the low-level assembly language implementation on the left.

ing the algorithm need not be too concerned about the details of the microprocessor-details which can be very distracting. This step allows the programmer to concentrate more on the details of the algorithm. This method is the familiar divide-and-conquer-approach, with the division taking place vertically, if you will, between the operand and comment fields on the assembly listing. See Figure 1.

Another advantage of this technique is the improvement of program documentation due to the built-in structure. Sometimes I have wished for a good bit of pseudocode to give me a bird's eye view of how someone else's assembly program works, rather than sighting down a hundred pages of the two black monoliths made of a seemingly endless column of code<tab>comments. This method provides that extra information quite naturally and displaces the flat comment-per-line documentation technique. In fact, the result is usually fewer but more meaningful program comments. Gone are pearls like, "add 5 to accumulator" on an add-immediate instruction, or, "is accumulator zero?" on a conditional branch

instruction. Such prattle is a waste of printer ribbons and owl habitat.

Many programmers use flowcharts faithfully when writing programs, maintaining them as the development of a program progresses. Though this is a good method, I believe the technique presented here is better for complex programs, because the documentation resides with the assembly code and is in structured form. Large flowcharts usually have mazes of interconnections which somewhat preclude block structure and encourage spaghetti code programming. However, augmentation of this technique with a flowchart im-

proves documentation as far as a general functional overview is concerned.

For the remainder of this article, I'll describe the details of the technique. It is presented as a series of steps.

## STEP 1.

Fully define the algorithm or task that you wish the computer to perform. This sounds obvious, but many programmers start typing mnemonics before they have finished defining the problem and its proposed solution, leading to spaghetti code and Bugs From Hell for other programmers to battle. Flowcharts, state transition diagrams, data flow diagrams, data structure maps, and so forth, are the net results of this step.

## STEP 2.

Choose a programming language in which you will express your algorithm. I suggest that it have all the constructs cited above for C, but that is not absolutely necessary. It is best to choose the structured language with which you are most familiar, although pseudocode (structured English) can be used.

### STEP 3.

Express your algorithm in the high-level language or pseudocode. To make the task easier and the program more readable, don't use any structures that you wouldn't want to code in assembly.

For example, I avoid ten-dimensional arrays and other similar beasts such as structures of arrays of structures, and so on. It is also best to express a complex mathematical operation as a series of simpler ones for clarity, with an eye toward using registers as temporary variables. Some data structure purists may wish to be more bold.

Indentation of program blocks is a key aid to program readability. And comment your high-level language code; it sounds ridiculous, but hard disk storage space on your PC is cheap—your time is not.

### STEP 4.

Once you have checked over your high-level language implementation and tested key algorithms by compiling and running them (if you have the high-level language compiler), you are ready to start assembly coding. Extract identifier names and their memory requirements from your algorithm and allocate space for variables in assembler. It is important that all major variables and data structures be defined and organized before you go to the next step.

### STEP 5.

Next, hand compile your high-level language program into the assembly language for your target machine. It sounds complicated, but is actually very easy. You don't have to know how the algorithm works to compile the program. The algorithm is contained in the high-level language version and need not be disturbed during the coding process. In fact, I have assembly coded programs which another programmer has pseudocoded with good results.

Start the compilation process by converting your pseudocode to assembly comments, typically by adding a few tabs and a comment character to the start of each line.

Listing 1—The IF-THEN-ELSE and IF-THEN constructs.

```
F- THEN- ELSE
      cmp    x,2      if (x == 2)
      jnz   L001      {
                          /* statement block */
L001:  jmp   L002      else
                          /* statement block */
L002:

IF- THEN
      cmp    x,2      if (x == 2)
      jnz   L001
                          /* statement block */
L00 . :
```

Then place labels on all branch destinations (see below for details). The form *Lnnn* is convenient, where *nnn* is a three-digit number. Adjust the label style to suit your needs and tastes. Descriptive labels are used only for subroutine and variable names. They are not needed in general because you can follow the indentation of the pseudocode to find branch destinations very quickly. This also eliminates the confusion of cryptic labels, meaningless to any but the original programmer, and then only for about ten minutes.

Next, add the assembly code. Each of the basic constructs mentioned above has a simple rule for compilation. The following examples use C as the high-level language. The mnemon-

ics used in my examples are those of the 80x86 microprocessor.

### IF-THEN-ELSE

See Listing 1. The conditional part of the statement is checked starting on the line containing the IF keyword. If the condition is not met, control is transferred to the line containing the ELSE keyword matched with that IF keyword. If the condition is met, the THEN code block executes and a jump instruction just before the ELSE keyword transfers control to the end of the IF-THEN-ELSE statement. The program falls out of the ELSE code block to the same point at the end of the IF-THEN-ELSE statement. Labels are required on the ELSE keyword and after the end of the ELSE block.

Listing 2—The WHILE construct.

```
L001  cmp    x,2      ;while (x == 2) &&
      jnz   L002      (y == 3))
      cmp    y,3
      jnz   L002
                          {
                          /* statement block */
      jmp   L001
L002
```

## WHILE

See Listing 2. The WHILE loop construct tests the continuation condition at the start of the loop. The loop can therefore execute zero or more times. A branch instruction at the end of the enclosed block of code always transfers control back to the top of the loop for testing of the continuation condition. The compound conditional tests of x and y in Listing 2 are coded separately for readability.

## DO-WHILE

See Listing 3. The DO-WHILE looping construct tests the continuation condition at the end of the loop. The loop executes at least one time. A branch instruction at the end of the enclosed block of code transfers control back to the top of the loop if the continuation condition is true.

## FOR

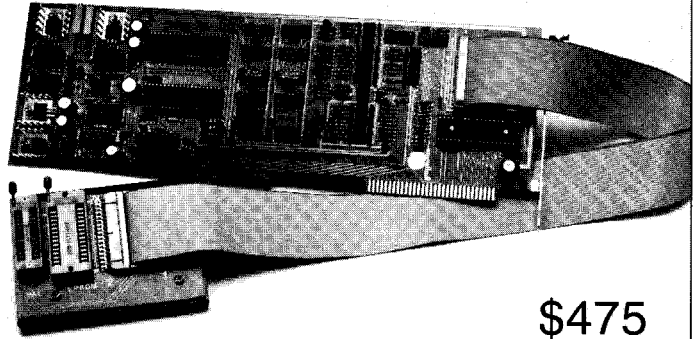
See Listing 4. The FOR construct is a special case of the WHILE construct where the loop continuation condition is derived from the value of a variable which is modified upon each pass through the loop. You can implement the more complex FOR conditions that C allows by changing the continuation test and INCREMENT instruction accordingly. Also, some processors have special decrement-jump instructions which can trim down the code size some.

## SWITCH

See Listing 5. The SWITCH construct allows the selective execution of one of a number of statement blocks depending on the run-time value of a single variable called the *control variable*. The `switch` statement consists of a sequence of comparisons of the control variable against constant values. Associated with each constant is a block of code to be executed when the control variable equals the constant. The comparisons occur in written sequence until a match is found, at which time the associated block of code is executed. Some C compilers search the cases bottom-up or use jump tables, but I like top-down searching for ease of

# Universal Device Programmer

PAL  
GAL  
EPROM  
EEPROM  
FLASH  
MICRO  
87C51  
PIC  
93C46  
XC1 736  
PSD 3xx  
5ns PALs

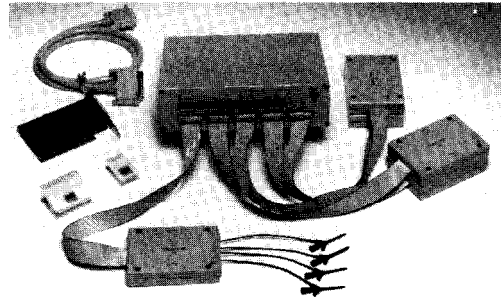


**\$475**

Free software updates on BBS  
Powerful menu driven software

## 400 MHz Logic Analyzer

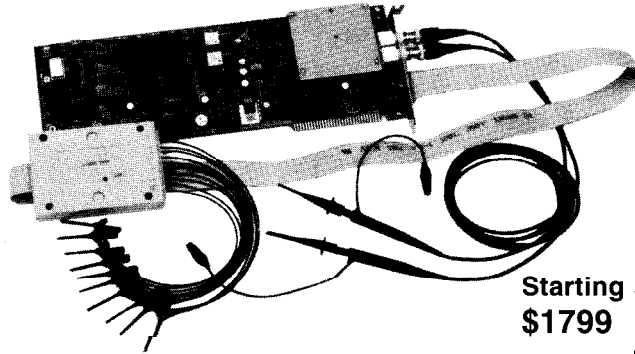
- up to 128 Channels
- up to 400 MHz
- up to 16K Samples/Channel
- Variable Threshold Levels
- 8 External Clocks
- 16 Level Triggering
- Pattern Generator Option



**\$799** - LA12100 (100 MHz, 24 Ch)  
**\$1299** - LA32200 (200 MHz, 32 Ch)  
**\$1899** - LA32400 (400 MHz, 32 Ch)  
**\$2750** - LA64400 (400 MHz, 64 Ch)

Price is Complete  
Pods and Software  
included

## 200 MSa/s DIGITAL OSCILLOSCOPE



Starting at  
**\$1799** With Pods  
& Software

- 200 MSa/s sampling Rate
- 2 Analog Channels (2 ch. Digital Oscilloscope)
- 8 Digital Channels (8 ch. Logic Analyzer)
- 125 MHz Single Shot Bandwidth
- up to 128K Samples/Channel

Call (201) 808-8990

Link Instruments



369 Passaic Ave, Suite 100, Fairfield, NJ 07004 fax: 808-8786

reading. If no match is found, a DEFAULT block of code is executed. Placing the most-likely executed cases at the top of the construct will speed the average execution time. And by all means, use jump tables if you need the speed.

## AN EXAMPLE

Let's look at an example. Say we need a routine to sort a buffer of 256 bytes in ascending order and sort time is not important. We are given the address of the buffer and permission to use any CPU registers. First, the buffer declaration looks like:

```
buffer ds 256
```

We assume it exists prior to the entry of our routine (it is global) and contains 256 bytes of unsorted data.

Since the data design has been done, we proceed to the rest of the problem: the design of the algorithm. A simple, double-loop, byte swapping scheme will do nicely. There. Now step I-algorithm design-is complete.

I have chosen C for the high-level language and the coded algorithm is in Listing 6. That completes step 2—choosing a programming language.

It took just a few minutes to scratch that out and type it in. I compiled this program with a Turbo C compiler and tested it with sample data to decrease debug time. With little effort, step S-coding the algorithm-is now complete.

Examining the code regarding variable space needs, I see three variables (*i, j, t emp*) which I can easily fit into CPU registers. This is a convenient situation because no memory space is needed for these local variables. Now step 4-data space allocation-is complete. The hand compiled version of the routine is shown in Listing 7.

You can see from this example that the indentation leads the eye quickly to branch sources and destinations, showing the basic structure of the program. Comments have been added indicating which registers are used for variable storage. This example program is ready to assemble and debug, completing step 5.

Listing 3—The DO-WHILE construct.

```
L001          ;do
              ; {
              /* statement block */
              ; }
              ;while ((x == 2) &&
cmp          x,2      ;while ((x == 2) &&
jnz         L002
cmp          y,3      (y == 3))
jz          L001
L002
```

## CONSIDERATIONS

Going further, procedures and functions are handled differently on each type of microprocessor, depending on the stack capability. Processors with a stack-indexed addressing mode make stack-based parameter passing easy and efficient. At the opposite end of the scale, control-oriented micros

with 4- or 8-level hardware stacks are not amenable to stack-based parameter passing and CPU registers or RAM must be used.

Due to these variations in architecture, uniformity in parameter passing is not possible as it is with the other control structures, so I will not consider it further here except to stress

Listing 4—The FOR construct.

```
1001:  mov     x,1      ; for (x=1; x!=10; x++)
      cmp     x,10
      jz      L002
      /* statement block */
      inc     x
      jmp     L001
L002:
```

Listing 5—The SWITCH construct.

```
      mov     ax,x      ;switch (x)
      cmp     ax,1      {
      jnz     L001      case 1:
                        /* statement block */
                        break;
L001:  cmp     ax,3      case 3:
      jnz     L002      /* statement block */
                        break;
L002:  cmp     ax,4      case 4:
      jnz     L003      /* statement block */
                        break;
L003:  cmp     ax,8      case 8:
      jnz     L004      /* statement block */
                        break;
L004:  jmp     L005      default:
                        /* statement block */
                        break;
L005:  ; }
```



the importance of consistency within a program. Comments at the start of each procedure or function should indicate values of input and output variables, how they are passed, and the operations performed on them.

One disadvantage of compiler-generated code is that it is usually not as efficient as that generated by an experienced programmer. The technique described here likewise introduces some code inefficiency. In deeply nested structures, one may find branch instructions that branch to branch instructions. But this adds little to program size and preserves the basic block structure that is one of the goals of this technique. However, memory restrictions may well dictate the elimination of every nonessential instruction in some situations. I have certainly had my back against the ROM wall and compromised in this area.

I have found program changes to be much simpler using this programming discipline. If a particular block must be rewritten, it is deleted, replaced with new pseudocode, and assembly coded. The block structure guarantees that there are no execution paths other than through the top and bottom of each block, so any block can be removed and coded again without worry of affecting another part of the program that might have used a piece of the deleted block. In short, the block structure eliminates any "spaghetti code" perils. The D-word (discipline) is very important here, for an ounce of spaghetti is worth a pound of curses.

Program porting is also aided with this programming technique. It is easy to use a word processor to strip the assembly code from a working program and recompile it for a different processor, either by hand or with an actual compiler. Most of the data structures will remain unchanged during the translation, too, speeding the task.

You will find that nested IF statements will produce multiple labels on one instruction. You might be tempted to remove all but one, but don't because the multiple labels make it easy to insert new code (another

**Listing 6—**The C coded sorting algorithm can be tested using a C compiler to eliminate logical errors before assembly coding. Remaining bugs in the assembly version are limited to the implementation, not the algorithm.

```
void sort void0

/* This function sorts a list of 256 bytes in ascending order.
The buffer[] is a global variable and is assumed to be filled
with data. A double loop swap sort is used. Register values are
undefined at entry and exit.
*/

int    i,          /* buffer index */
       j,          /* buffer index */
char   temp        /* used in byte swapping */

/* double loop compares each pair of bytes */
for (i=0; i<255; i++)

    for (j=i+1; j<255; j++)
    {
        if (buffer[i]>buffer[j])
        { /* compare pairs of bytes */
            temp=buffer[i]; /* swap the bytes */
            buffer[i]=buffer[j];
            buffer[j]=temp;
        }
    }

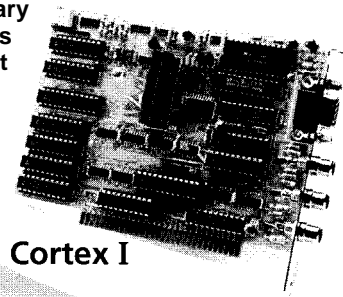
/* end of sort0 */
```

## Video Frame Grabber

- \$495 Including Software with "C" Library
- Half Slot Card for Compact Applications
- Real Time Imaging with Display Output
- 8 Bit (256 Gray Levels)

### FEATURES:

- Single 512 X 484 Image or Four 256 X 242 Images
- External Trigger
- Input Look Up Table
- Low Power Option Available
- Elegant Software Interface
- <10 nsec Pixel Jitter Means Accurate Digitization
- EISA (PC) Bus and STD Bus Products Available
- RS-170 and CCIR Video Formats Available
- Binary and TIFF File Formats



Cortex I

The Cortex I is used in machine vision, industrial control, medical, security and scientific applications around the world.

ImageNation strives to delight customers with quality products and personal service at a competitive price.

**Call today for volume pricing or to discuss your application.**

**ImageNation Corporation**  
Providing Imaging Solutions  
FAX (503) 643-2458 • (800) 366-9131

P.O. Box 276  
Beaverton, OR 97075  
(503) 641-7408

ELSE clause, for example) at a later date without affecting other branch destinations. Remember, labels are cheaper than your time.

You may also use assembler macros to ease the coding process, though the examples in this article do not.

Since there is no external authority imposing discipline on the programmer using this system (as a compiler would), each programmer will use it a little differently. This is okay. The main goal is to foster a consistency of style and structure, first to increase productivity, and second to allow following programmers to understand the subject programs.

In conclusion, the advantages of this method are the separation of the algorithm from the assembly language, improvement of documentation through the use of structured comments, and improved mental health of your successors. The first item saves time now, the second later, and the third may save your life.

I have used this discipline to write numerous assembly language programs for ten years on 4-, 8-, and 16-bit machines with considerable success and positive comments from other programmers. I adopted the technique after writing programs using other methods and have noticed a dramatic increase in productivity, ease of program development, and consistency of documentation. I hope this method allows you similar profit. ☺

Hank Wallace is the owner of Atlantic Quality Design Inc., a consulting firm located in Rural Hall, N.C. He has designed embedded system software and hardware in the areas of data collection, machine control, microprocessor emulation, satellite and mobile radio communications, radiopaging, manufacturing test, and speech privacy.

Listing 7-Assembly coded sorting algorithm. Indentation reveals the block structure of the program and allows quick location of branch destinations. For porting to another processor, the assembly code can be stripped and the program recompiled, either by hand or by computer.

```

;void sort(void)

; /* This function sorts a list of 256 bytes in ascending
; order. The buffer[] is a global variable and is assumed
; to be filled with data. A double loop swap sort is used.
; Register values are undefined at entry and exit. */
; {

; int
;     i,          /* buffer index, register SI */
;     j;          /* buffer index, register DI */
; char temp;     /* used in byte swapping, register AH */

.model small
.data
extrn  buffer

.code
sort:
    mov     ax,seg buffer
    mov     ds,ax
    mov     bx,offset buffer

; /* double loop compares each pair of bytes */
L001:   mov     si,0           ; for (i=0; i<255; i++)
        cmp     si,255
        jnb   L006
        mov     di,si
        inc    di           ; for (j=i+1; j<256; j++)
L005:   cmp     di,256
        jnb   L002
        mov     al,[si+bx]   ; if (buffer[i]>buffer[j])
                                ; /* compare pairs of bytes */
        cmp     al,[di+bx]
        jbe   L004
        mov     ah,al
                                ; {
                                ;     temp=buffer[i]
                                ;     /* swap the bytes */
                                ;     buffer[i]=buffer[j];
                                ;     buffer[j]=temp;
                                ;
L004:   inc    di
        jmp   L005
L002:   inc    si
        jmp   L001
L006:   ret
                                ; } /* end of sort.0 */
end

```

## IRS

410 Very Useful

411 Moderately Useful

412 Not Useful

# DEPARTMENTS

48 Firmware Furnace

58 From the Bench

64 Silicon Update

70 Embedded Techniques

77 Patent Talk

83 ConnecTime

## FIRMWARE FURNACE

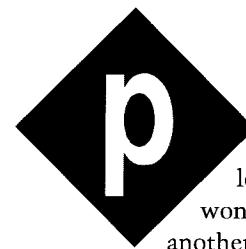
Ed Nisley

### Hardware for the '386SX Project's Bitmapped LCD Panel



On with  
the show.  
Ed  
proceeds

with the addition of a Graphic LCD Interface to his Firmware Development Board. It's not all pixels and dots from here; instead it is timing and control that must be dealt with.



Perhaps we've lost our sense of wonder: name

another manufactured product with a quarter of a million parts laid out for your inspection. Every dot on that 640x400 LCD panel must be essentially perfect because human eyes are very, very good at picking out tiny differences. Sure, a CD-ROM has more dots, but it's just a silver disk to anything but an electron microscope.

This month I will cover the hardware that puts every one of those dots under your control. I'll concentrate on 200- and 400-line displays because they're readily available, but you'll also see some of the changes needed for other panels. The block diagram in the January column will guide you through the schematics.

In March, I'll cover the test code in more detail, but you can download the program files from the BBS right now.

#### COUNTING CLOCKS

The heart of the Graphic LCD Interface is the LCD Timing and Control section. It generates the clock and sync pulses required by the LCD panel, as well as the control signals that orchestrate the rest of the interface. Figure 1 is the schematic for this part of the circuit.

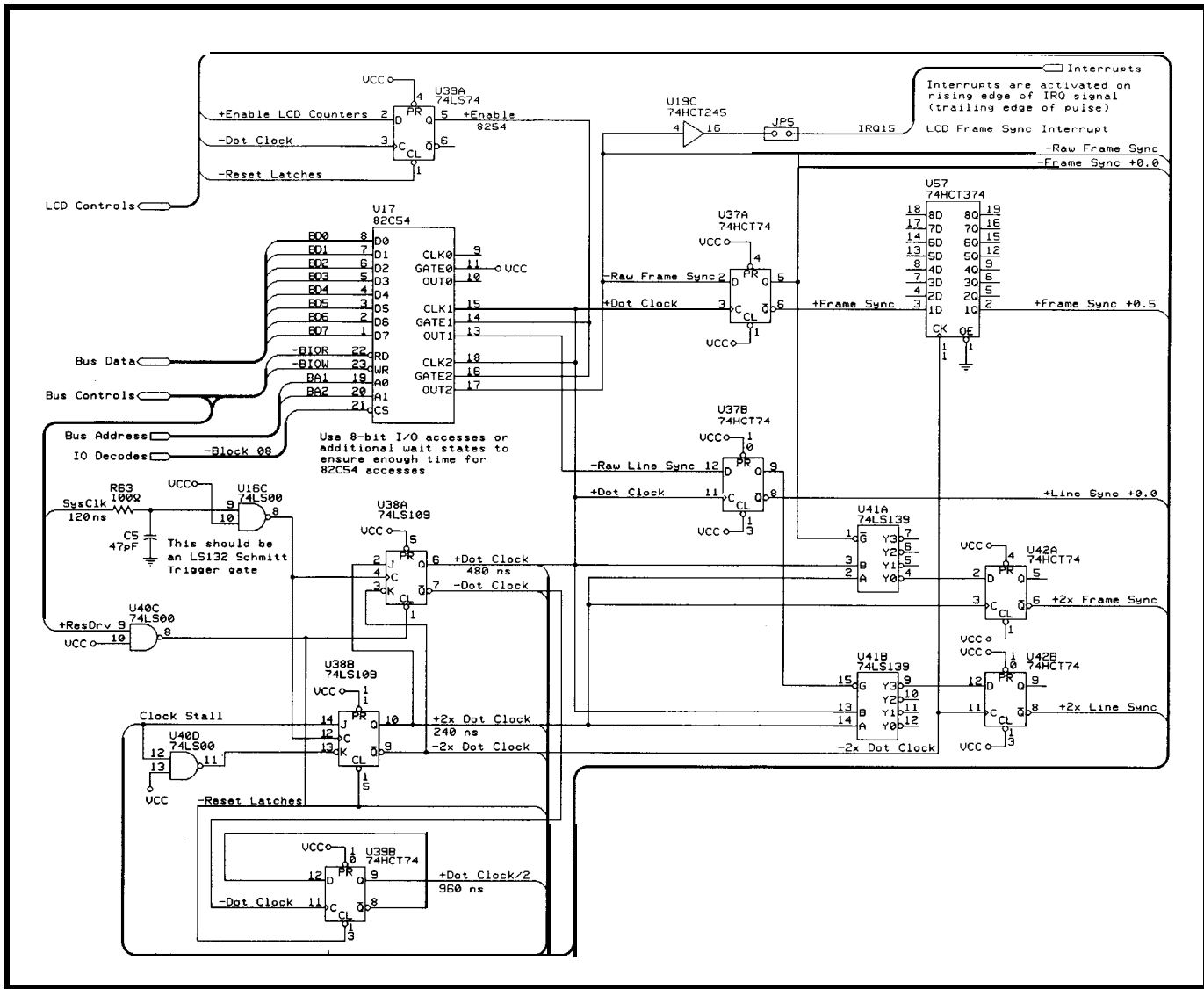


Figure 1—This circuitry defines the fundamental Dot Clock period that drives the rest of the Graphic LCD Interface. The 82C54 was covered in a previous column, but its clock and gate lines have changed. The various Dot Clock and Sync signal outputs can control a variety of panels.

The ISA bus SysClk signal defines the basic interface timing. U38, an 'LS109 dual J-K flip-flop, divides the 120-ns SysClk by two to produce the double-speed 2xDot Clock and the regular 480-ns Dot Clock. U39b, part of an 'HCT74, produces the half-speed 960-ns Dot Clock needed for slower panels.

Back in Issue 34 (May 1993) we got the Firmware Development Board's 82C54 working. I said then, ".I may come up with other applications that require a wiring change.." and you'll be pleased to know that this is just such an occasion. U17 in Figure 1 is the 82C54 you've already checked out, with two channels now devoted to metering out Dot Clock cycles.

Counter 1 produces the -Raw Line Sync signal by counting the number of 480-ns Dot Clocks in each LCD row. The 640x200 and 640x400 panels I have require either 160 or 320 clocks per row, depending on whether they have a single- or double-speed interface. Other panels have different requirements, so it made sense to use a programmable counter rather than a bunch of jumpers and gates.

Yes, I know it's inconsistent to talk about LCD rows and Line Sync signals. That's the way my notes and sketches started out.. .and it's too late to change now!

Counter 2 controls the number of Dot Clock cycles in one LCD display frame by generating the -Raw Frame Sync signal. Both 200- and 400-line

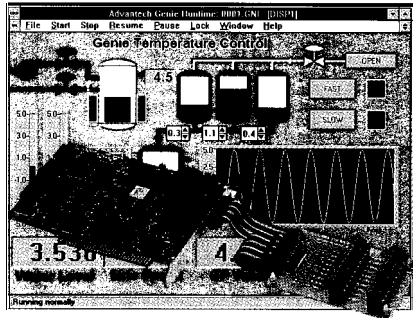
panels require 32,000 clocks, but other panels need different counts. Here, the 82C54 serves as a compact 16-bit programmable counter.

Figure 2 is a diagram timing for the key Graphic LCD Interface signals. The 82C54 timing specs say the output delay may be up to 100 ns after the falling Clock edge, so you can't use the signals for critical timings. I use an 'HCT74 to synchronize the edges with +Dot Clock; the resulting pulses occur at the "+0.0" times relative to the rest of the signals.

Both channels must start counting on the same Dot Clock cycle so that the two sync signals have the right relationship. U39a disables the counters until the firmware sets the +Enable LCD Counters bit in the LCD

# Data Acquisition & Control Solutions

Temperature Measurement  
High Speed Data Streaming  
Remote Data Acquisition  
Strip Chart Recording



Temperature Measurement & Control Solution Package ~ \$895

## Key Features:

- ❖ For J, K, S, T, B, R, E thermocouples
- ❖ 8 thermocouple channels, expandable to 32 channels
- ❖ 0.1°C resolution with 12-bit A/D
- ❖ Up to 250 samples per second
- ❖ 16 D/I and 16 D/O channels
- ❖ Powerful Windows SCADA software
- ❖ Icon based, no programming required

1 General Purpose DAS Card 8 A/D, 1 D/A, 16/16 DIO, wiring kit	\$295
1 Multifunction DAS Card 16 A/D, 2 D/A, 32 DIO, counter	\$395
1 High Performance DAS Card 100KHz, programmable gain	\$595
1 24 Ch Digital I/O Card	\$100
1 24 Ch Digital Input w/Interrupt	\$180
7 32 Ch Digital I/O Card	\$170
7 32 Ch Digital Input w/Interrupt	\$235
7 144 Ch Digital I/O Card	\$295
1 8 Relay & 8 Digital Input Card	\$210
1 High Speed Data Streaming Pkg. 100KHz data streaming to disk	\$595
1 PC Strip Chart Recorder Pkg. 16 channel recording at 15KHz	\$695
7 Remote DA&C Starter Kit RS-485 based, up to 4000 feet	\$495

Free  
120-page Solution Guide  
for quality minded,  
budget conscious  
Engineers



1-800-800-6889

Industrial & Lab Automation with PCs  
**ADVANTECH**

750 East Arques Ave. Sunnyvale, CA 94086  
Tel: (408) 245-6678 FAX: (408) 245.8268

Control port. When U39a's clock rises, both 82C54 Gate inputs go to a high state and the counters are off and running.

While I was debugging this circuitry, the LCD panel would occasionally go blank. It turned out that Counter 1 stopped counting unpredictably for periods between a few cycles and a few hundred milliseconds. This immediately threw off the critical timing between Line and Frame Sync. The 82C54 Gate input remained high and Dot Clock continued to run during the stoppages.

The 82C54 has one of those classic "you must read every word" data sheets. The Clock input timing specs include this tiny footnote: "Low-going glitches that violate  $t_{PWH}$ ,  $t_{PWL}$  may cause errors requiring counter programming." Translated into English, this says that if the clock isn't perfect the counter will misbehave in strange and mysterious ways. That footnote accounts for the RC filter and gate driving U38.

We think of TTL logic devices as digital: their outputs are either high or low, never anywhere in between. In reality they are analog circuits with a nonlinear transfer characteristic. If the input voltage is between the valid logic 1 and logic 0 levels you *will* get something out, but it *won't* be a Boolean value.

Dot Clock was not the perfect square wave I expected. Once in a while the high half-cycle was dented by a low-going glitch. After some delicate probing I discovered an impressively noisy SysClk signal on the backplane bus. The clock noise was bad enough to force the 'LS109 latches into metastable states and the ensuing invalid outputs stunned the 82C54 counters. While they didn't need reprogramming, they sure didn't work right!

The correct solution for this problem is a Schmitt trigger gate in the clock path. Unlike a standard logic gate, a Schmitt trigger has analog feedback that modifies the input switching levels. As a result, most glitches simply Go Away. But I didn't want to add an 'LS14 just to get one inverter.

So I used a simple RC filter to knock off the noise and a spare 'LS00 gate to square up the result. Yes, I know I could have substituted an 'LS132 (a Schmitt trigger NAND), but the other circuits using that package couldn't stand the additional delay. An RC filter is not a textbook solution, but it works for my application. Take a look at your SysClk, then tell me what you see and how you attack the problem.

Rest assured it took me *much* longer to discover and solve that problem than to write about it!

Although the 82C54 defines the overall row and frame timing, its output pulses cannot drive the LCD panel directly. The dot data and sync signals must both arrive with the correct polarity at the proper times, so we now consider the dots.

## A CHAIN OF DOTS

A 32K-byte SRAM holds the binary data that will become dots on the panel. The RAM, U45 in Figure 3, is time-shared between the ISA bus and the LCD panel. Obviously only the PC can write to it, but both circuits can read the dot data. The Dot Clock signal determines which circuits have access to the RAM at any given time.

The LCD Address Counters, U43 and U44, provide the 15-bit address for each RAM byte when the LCD is in control. Each 'LS590 chip contains an S-bit counter followed by an 8-bit latch with tristate outputs. The counter and latch are both driven by -Dot Clock, so they change state on the falling edge of +Dot Clock. In this application, the Address Latch is always "one behind" the Address Counter because it captures the count just before the clock edge.

Frame Sync clears the counters to zero at the beginning of each panel refresh. The actual reset signal occurs during only a quarter of the Dot Clock period to ensure that the counter will step from 0 to 1 on the next clock pulse. Figure 2 shows the timing: the reset pulse must be finished by the next -Dot Clock rising edge

The 'LS590 Address Latch solves a critical timing problem: the path

length from the Frame Sync reset logic through the counters and RAM into the LCD Data Latch is longer than 240 ns. Latching the address provides a stable signal for the entire LCD half of Dot Clock, while delaying the RAM's output data by one cycle.

The RAM address alternates between the Address Latch and the address on the PC's ISA bus connector. The 'LS590 outputs are active when +Dot Clock is low, while U46 and U47, the PC Address Buffers, are active when -Dot Clock is low. Figure 2 shows the sequence of events; note how the LCD Address Latch ensures that the address remains valid while the counter is being reset.

The LCD Data Latch, U48, captures the RAM output when Dot Clock goes high 240 ns after the LCD's half cycle begins. This ensures stable data at the LCD panel when Dot Clock goes low 240 ns later. The panels have various timing specs, with some requiring at least 150 ns of setup and hold time. Without this latch, the data at the LCD panel would disappear long before the panel was done capturing it.

### OFF BY ONE...

Pop quiz: (1) According to the LCD panel specs, when does the Frame Sync pulse occur in relation to the first row of data? (2) In Figure 2, what is the RAM address of the value in the Data Latch when Frame Sync is active? (3) How do you resolve this paradox!

Answers: (1) After 160 Dot Clocks, at the start of the *second* row of panel data. (2) 31,999, plus or minus one depending on how you read the diagram. (3) Ah, Grasshopper, consider the sound of one hand clapping.. forehead.

Figure 4 is the Optrex DMF651 640x200 panel pulse timing diagram, redrawn using my nomenclature. There are three vital synchronizing relationships: Line Sync/Dot Clock, Frame Sync/Line Sync, and Alternate Frame/Line Sync.

The falling edge of Line Sync must be more than 80 ns away from a falling Dot Clock edge. U37b delays the 82C54's -Raw Line Sync output until

the next rising Dot Clock. Figure 2 shows that +Line Sync+O.O rises when the last data byte in each row appears in the Data Latch and falls on the first byte of the next row.

Frame Sync must be high for at least 100 ns on either side of the first Line Sync's falling edge. U37a synchronizes -Raw Frame Sync to the next rising Dot Clock and U57 delays it by half a Dot Clock cycle. Figure 2 shows the result: the pulse called +FrameSync+0.5 is neatly centered on the falling edge of +Line Sync+O.O with about 240 ns setup and hold times.

The Alternate Frame signal comes from U52 in Figure 5. For the DMF65 1, this signal must change state within 300 ns of a Dot Clock falling edge, so I clocked U52 with +FrameSync+0.5. The Hitachi LM215 refers Alternate Frame to the falling edge of Line Sync rather than Dot Clock, so a jumper allows you to use -Frame Sync+O.O instead.

U41 and U42 generate the sync signals needed for panels that accept data every 240 ns. The double-speed 2xDot Clock changes the allowable width and alignment of the sync signals, but the basic Graphic LCD Interface hardware remains the same. The firmware sets up the LCD Data Multiplexer to switch between the two nybbles every half Dot Clock cycle, so the data arrives on time. I'll go into more detail later.

You must drive all LCD signals with CMOS logic gates to meet the panel's minimum  $V_{IH}$  spec. Because I used several spare gates from other sections of the FDB's logic, some of the LS gates mentioned in previous columns are now HCT. In actual practice, LS gates will probably work, but only by the grace of good tolerances.. which is bad practice.

### PORTING A RAM

When Dot Clock is high, the PC controls the Graphic LCD Interface RAM. U46 and 47, the PC Address Buffers, are enabled during that half-cycle. U49, another 'LS245, shuttles PC data to or from the RAM. These buffers are in series with those already on the FDB.

# \$149 ICE?!!!

Yes, that's right! HTE has dropped the price of it's popular 8031132 Enhanced **DryICE** from \$269 to \$149. Now you can't afford to be without one! This ICE performs single step, real-time execute to breakpoint, disassembly and more. **8K** user code space, expandable to 32K. Order yours today!

## More ICE

Our **DryICE Plus** has so many features (the price is unbelievable. 48K of user code space, real-time execution, and expandability to nearly all of the 8051 family processors. We are now supporting pods for the 8031/2, 8751, 80CL51/32, 8051Fx, 80C154, 80C410/1, 80C451, 80C528, 80C550, 80C535, 80C537, 80C552/562, 80C575, 80C652, 80CL781/2, and 80C851. With pods priced at only \$149, going to a different, higher integration processor is easy. Pod and base unit are just \$448 complete!

## Controller 80C552

At \$149, our **552SBC-10** OEM board has the price and features you need right now! It's an 8051 core processor with an eight channel, 10-bit A/D, two PWM outputs, capture/compare registers, 16 digital I/O, one RS232 serial port, four JEDEC memory sockets. Add options like two more RS232/422/485 ports, 24 more digital I/O, real-time clock, serial EEPROM, and battery-backup for clock and RAM. Start with the Development board; all the peripherals and a debug monitor for only \$349. Download and debug your code, assembly or 'C', right on the SBC, then use the low-cost OEM board of your choice for production. We also do customs - call for a free quotation.

## 80C188 SBC

NEW! Starting at only \$249. Two serial RS232/422/485, Parallel, expand to 16 ch. 12-bit A/D, 8 ch. 12 bit D/A, Keybd, LCD, Relay I/F, more. Call for details!

Call for your custom product needs.  
Quick Response

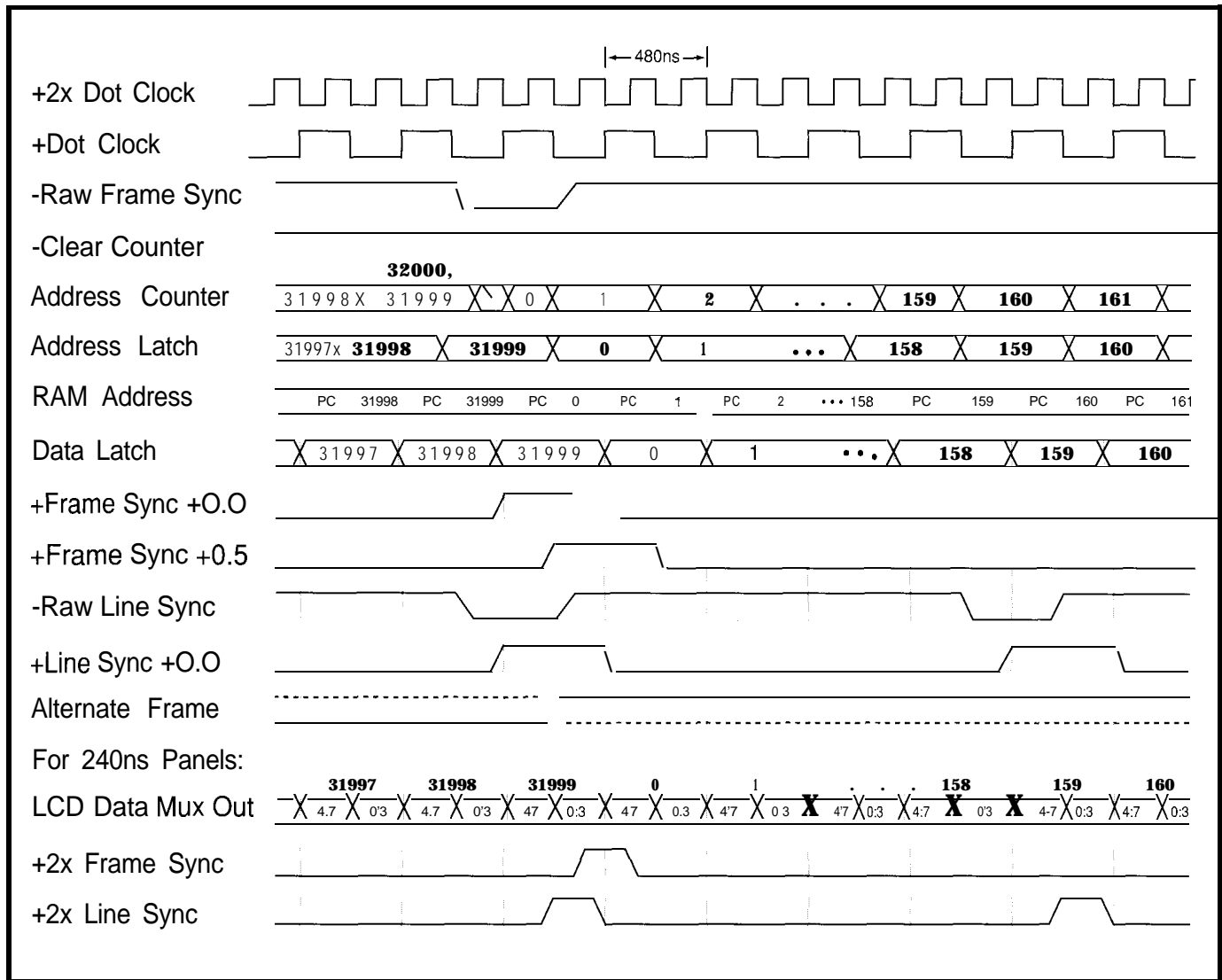


# HTE

HiTech Equipment Corp  
9400 Activity Road  
San Diego, CA 92126  
[FAX: (619) 530-1458]

(619) 566-1 892

70662. 1241 @compuserve.com



**Figure 2**—This **timing diagram** summarizes the Graphic LCD Interface's operation. The 32-Kbyte RAM holding the LCD dot patterns is time-shared between the PC ISA bus and the LCD circuitry. The -Clear Counter signal resets the LCD Address Counter at the end of each frame of data; that signal originates in the 82C54 counter described in an earlier column.

There is one catch in the PC's access: the Dot Clock cycle is not synchronized with the ISA bus read and write cycles. If it goes low at the end of a bus read, the CPU will get invalid data or write junk into the RAM because the PC's buffers are turned off and the LCD circuitry is in control.

Solving this problem means the PC must wait until Dot Clock goes high or the LCD panel must wait until the PC is finished. I chose the latter course because you can pause an LCD clock for a very short time without visible effect...and the PC access is slow enough already. You can do it the other way if you like.

Photo 1 shows how this works. The top trace is -SMemW, which

controls the ISA write cycle timing. The lower trace is Dot Clock, which is normally a 480-ns square wave. Notice that it's always high when -SMemW rises, even though it's not synchronized when the cycle begins. The normal clock cycle resumes after -SMemW ends.

A complete 8-bit ISA bus memory write is six SysClk cycles long, with -SMemW active for about the last 4.5 cycles. Because Dot Clock is four SysClk cycles long, the maximum delay until Dot Clock goes high after -SMemW becomes active is three SysClk periods.

U56a monitors the double-speed 2xDot Clock, Dot Clock, and the (negative active) OR of -SMemW and -SMemR. When Dot Clock goes high

during a PC access, U56a's output freezes the SysClk divider flip-flops and holds them until the PC is finished.

The shortest RAM access is thus about 1.5 SysClks. Allowing for the buffer delays, a 120-ns RAM is just about fast enough. I suspect some systems may need a 100-ns RAM or 74F buffers instead of 74LS parts. In any event, work out your system's timings and see how close I am.

At worst you can add a wait state or two to the PC access. Don't make the mistake of stalling both Dot Clock and the PC, though; if both sides of the interface are stopped, neither one will restart!

Even though a Dot Clock cycle can be stretched by up to 3SysClks

when it collides with a PC access, as long as this is relatively infrequent you'll never see it. One of my test routines writes data into the RAM while toggling a parallel port bit to provide scope sync: one byte every 33  $\mu$ s affects an invisible 1.5% of the Dot Clock cycles.

On the other hand, another test routine has a REP LODSB in a loop that reads all 32K bytes at 960 ns/byte. That severely distorts the Dot Clock cycle, holding it high for 720 ns and allowing only one 240-ns low half-cycle between each access. The resulting refresh rate drops to 33 Hz and the panel develops an obvious flicker.

But remember, even if you used REP MOVSB to read or write all 32K bytes, you wouldn't do it a loop. The LCD panel's optical response is in the 100-200-ms range, so it wouldn't have time to respond to 30 ms of furious writes; the new dots would take another hundred milliseconds or so to replace the old ones! Stop worrying... it works OK.

Do note, however, that this interface isn't suited for DMA or bus-mastering card access because the timings depend on the relation between SysClk and -SMemR/W, which is valid only for system board accesses. If you need a more versatile board, check Solari's book for the grim details.

Even when all your logic is working correctly, you won't see anything on the LCD panel unless you apply the right drive voltages. I won't mention how often I've caught palpitations at the sign of a blank panel only to realize the power supply lead was dangling.

### PRACTICAL POWER

The power supply circuitry that is shown in Figure 5 is overkill. You really only need perhaps 20 or 30 mA at a single voltage, so there's no need for a bulky wall-wart transformer, a huge capacitor, and a 1.5-amp, 15-watt voltage regulator in a TO220 case. Suffice it to say there are easier and more compact ways to generate drive voltages for an individual LCD panel.

But if you're working with a variety of panels, it makes sense to use parts from your stash for a versatile supply. Maybe one of these days we can convince Jeff to do a "From The Bench" column on LCD power

supplies? Maybe he'll even include backlight inverters!

K1 is my concession to the power sequencing specifications found in the data sheets. Most of the panels are happy if you apply +5 V first and

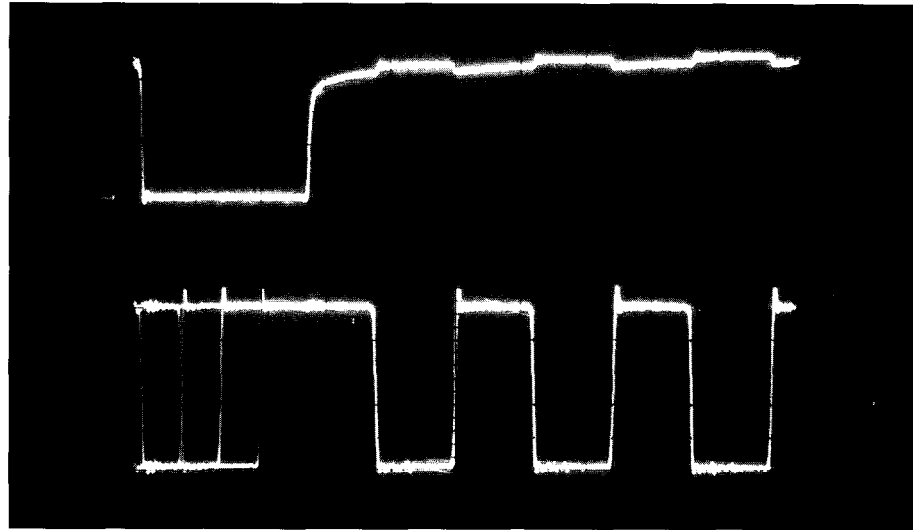
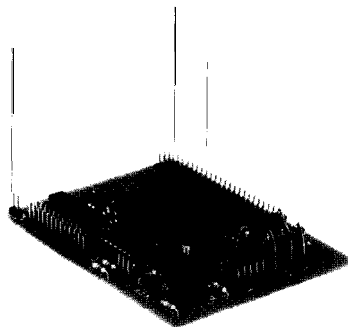


Photo 1--An ISA bus read or write access forces the Dot Clock to stall in the high state until the operation is complete. The top trace, -SMemW, is active for about 4.5 SysClk cycles. Dot Clock, shown in the bottom trace, is not synchronized to -SMemW when the write cycle begins, so it may stop at any of the four times shown in this multiple exposure. The RAM access starts when Dot Clock goes high and ends when -SMem W goes high, which is just enough time for a 120-ns RAM cycle.

## Drop 68HC11 Flexibility into Your Application



**Don't worry about which microcontroller to choose when you can rely on the industry leader.**

**-MODULAR, STACKABLE DESIGN ENSURES SMALL FOOTPRINT AND BIG PERFORMANCE.**

**-ADD ONLY THE FEATURES YOU NEED - LOWERING OVERHEAD**

**-FULL FEATURE 'C' COMPILER AND MACRO ASSEMBLER AVAILABLE**

**- PRICES START AT \$59 IN SINGLE QUANTITIES**

### Micro Wonders

320 Brown St Suite 404

West Lafayette, IN 47906

(317) 420 - 1686



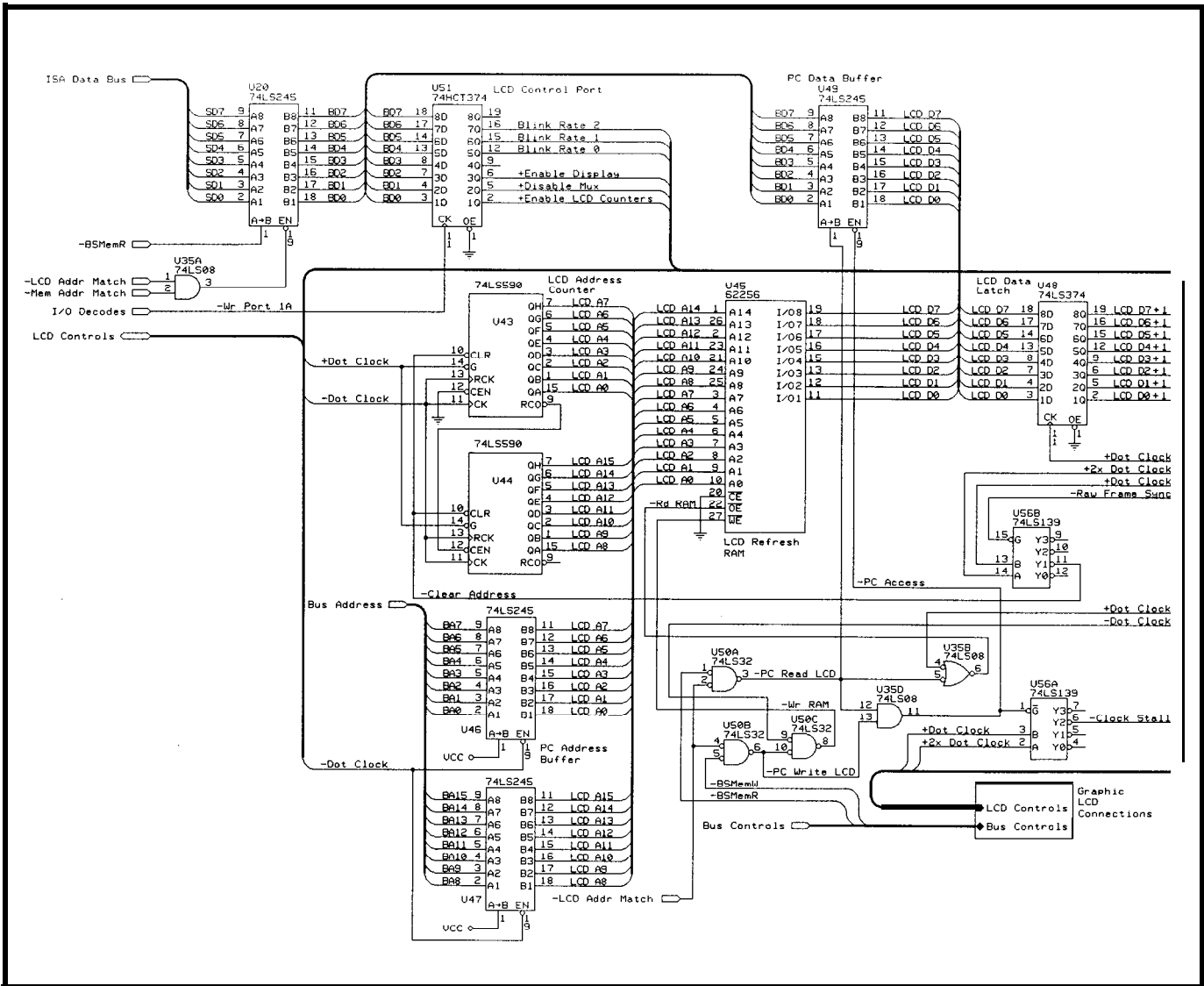
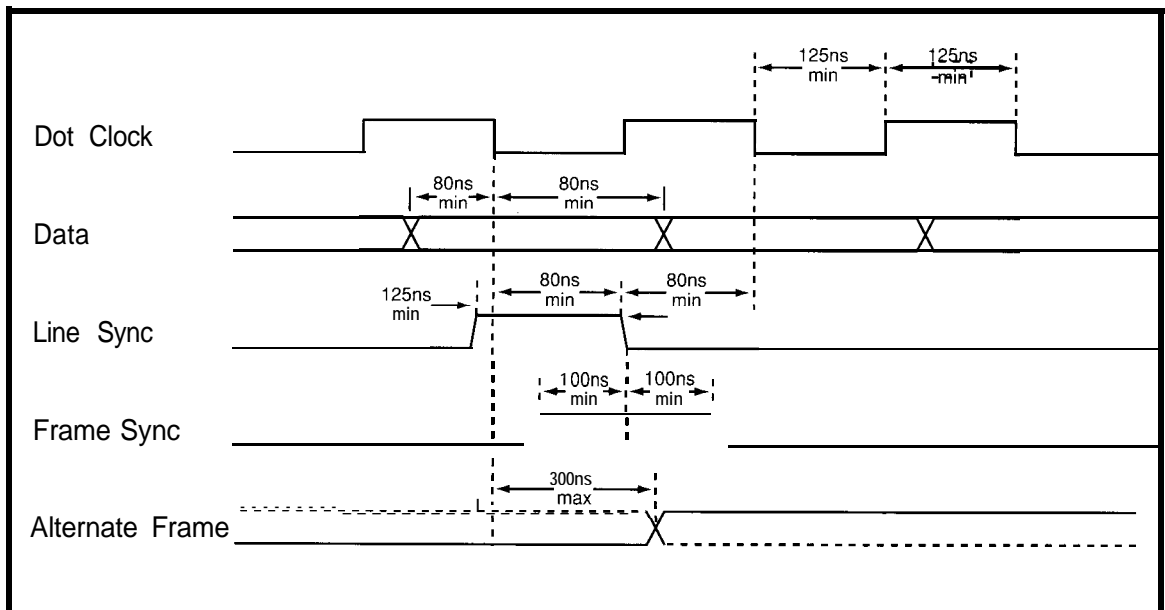


Figure 3—The RAM may be accessed by both the ISA bus and the LCD Address Counters. The -LCD Address Match signal is produced by a 74F521 comparator similar to the one shown in Issue 36. The Dot Clock signal controls the three-state buffers driving the RAM's address lines. The 'LS139 decoder generates two signals that depend on the Dot Clock phase to control other parts of the circuit.

Figure 4—The Oprex DMF651 signal timing specifications dictate how the Graphic LCD Interface must operate. Notice that the data and Line Sync must be valid on the falling edge of each Dot Clock pulse, but Frame Sync is related to the falling edge of Line Sync. Other panels have different specs, but the Graphic LCD Interface can handle many of them with a few jumper changes.



remove it last, which is what the DPDT relay accomplishes by disconnecting the LCD drive voltages. The ISA bus +RstDrv signal is active when the power supplies are out of tolerance or the system is reset, which ensures +5 V is always active first and last.

**CAUTION:** set the LCD drive voltage *before* you connect the panel, then monitor a voltmeter when you adjust it. The supply shown in Figure 5 can apply enough voltage and current to destroy a panel with a twist of a screwdriver. Be hypercautious here, if nowhere else!

## GREATER&LESSERARRAYS

The circuitry so far can handle panels with 480-ns Dot Clocks. The situation is slightly different for panels needing 240-ns or 960-ns clocks, but the hardware modifications are easy. My board is festooned with jumpers, but you can build only the parts needed for your panel.

The Sharp LM64015T 640x400 panel expects 320 Dot Clocks on each of 200 rows with a 240-ns Dot Clock. The RAM still delivers eight bits every 480 ns, so the firmware sets up the LCD Data multiplexer to switch between the two nybbles on each half Dot Clock cycle. As a result, the panel's dots line up quite nicely with the data bytes: the high and low nybbles are contiguous at last!

The Frame and Line Sync pulses are 240 ns long and synchronized with the double-speed 2xDot Clock. I used U41, an 'LS139 decoder, to control the two flip-flops in U42 with the result shown in Figure 2. The panel is driven by -2xDot Clock to get a falling edge in the middle of each nybble.

The Sharp LM641481 640x480 panel is similar to the LM64015T except that the extra 40 (not 80!) rows reduce the refresh rate to about 54 Hz. This is another high-performance panel with a brilliant cold cathode fluorescent tube backlight. Unfortunately, to my eyes that 54-Hz refresh flicker is quite obvious. As with the LM64015T, the CCFT inverter is really hard to find.

A 640x480 panel requires 38,400 bytes of RAM, which the standard

8051

68HC11

68HC05

COP8



# THE INCREDIBLE NEW PE-8351 FX IN-CIRCUIT EMULATOR **ONLY \$1451.00**

- supports 8X51/31, 8X52/32, 8X51FA/FB/FC
- Real-time and Nonintrusive
- 64K Program Memory
- 64K Data Memory
- 128K Hardware Breakpoints
- 16K Frame Trace Buffer
- Transparent Trace (View Trace and Execute Simultaneously)
- Fast, Easy Installation to RS232 Port of any Dos PC, even Laptops!
- Built-In Self-Test
- Symbolic Debug
- Source-Level Debug

- Other Fine Emulators**
- 8051 From \$1400\*
  - 68HC11 From \$1600\*
  - 68HC05 From \$1700\*
  - COP8 From \$625\*
  - More than 100 devices supported through interchangeable probe cards

- New Debugger Enhancements**
- Full support for structures, unions, arrays, pointers
  - Data structure browser/editor
  - True expression in watch window-detect bad pointers

Nobody Matches the Value of the PE Family

PE Product	Max Freq.	MetaLink List Price*	NOHAU List Price*+
8351 FX	16	\$1451	\$4820
837511752	16	851	4300
8032-24	24	851	4945
8032-42	42	999	7195

\*U.S. Retail price. \*U.S. Price List dated 3/92, verbal quote 8/93.

### METALINK INVENTED LOW-COST EMULATION IN 1984.

Our new AET Emulator architecture (Advanced Emulator Technology, Pat. Pending) provides unmatched value. MetaLink also delivers leading edge customer service, including a 30-day money back guarantee, 10 day trial periods, rental plans and free technical support. Call today for FREE demo diskette.



(800) 638-2423

**MetaLink**<sup>®</sup>  
Corporation

MetaLink Corporation  
325 E. Elliot Road, Chandler, Az 85225  
Phone: (602) 926-0797 Fax: (602) 926-1198

Metatink Europe GmbH  
Westring 2, 8011<85614> Kirchseeon-Eglharting  
Teleton (08091) 2046. Teletax (08091) 2386

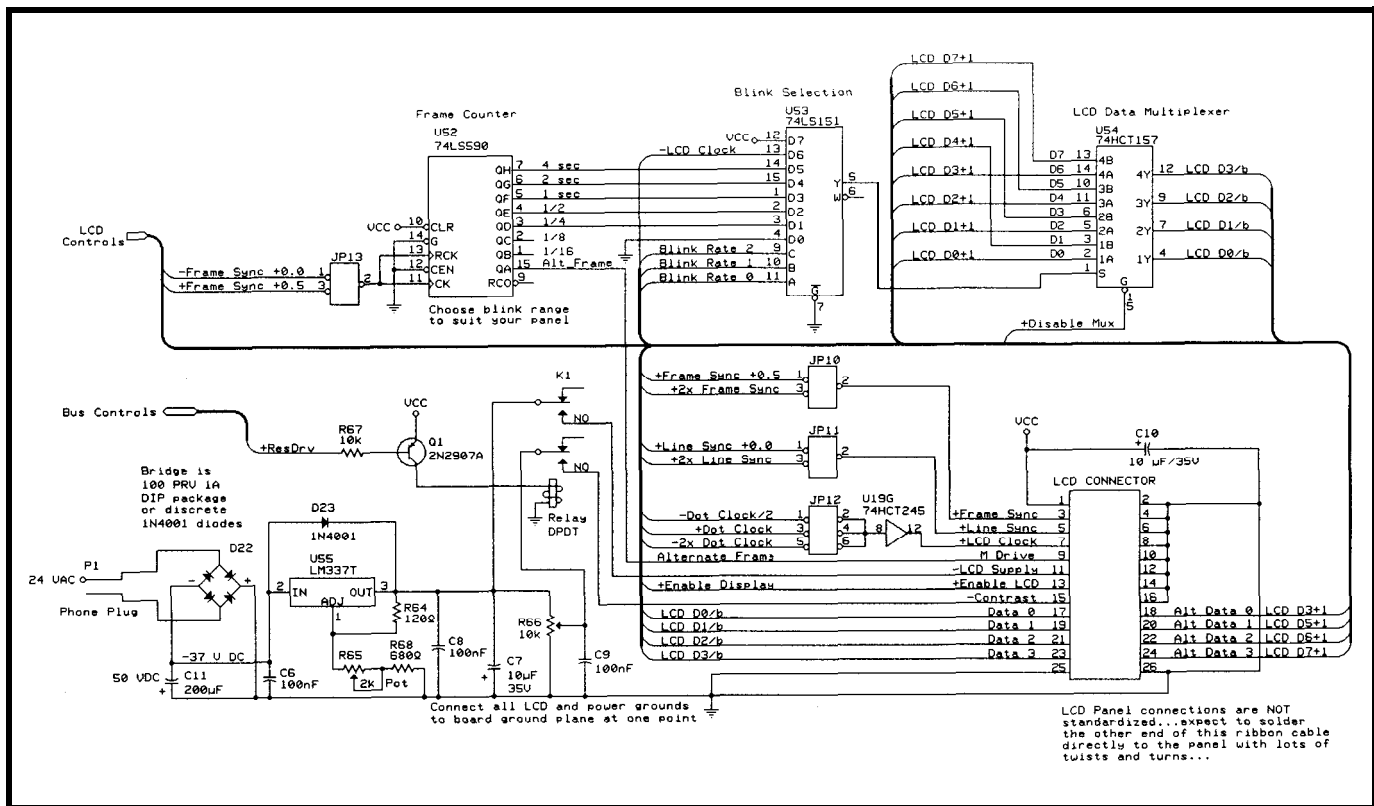


Figure 5—Because each LCD panel has different requirements, this part of the schematic is festooned with jumpers. If you build this circuit to drive a particular panel you can hardwire the signals. U53 can select a signal that alternates U54 between the nybbles of each byte to blink the panel if it uses 4-bit data and a 480-ns clock. The power supply can drive nearly any common LCD panel, but must be adjusted carefully to stay within the panel's voltage specifications.

Graphic LCD Interface just doesn't have. Although it's not shown on the schematic, you can piggyback two 32K RAMs with their -CE inputs driven by +LCD A15 from U44 and -LCD A15 from an inverter. Disconnect ISA bus address line A15 from the 'F521 so you can access all 64K bytes between D000:0000 and D000:FFFF.

In contrast to that vast array of dots, the Hitachi LM215 480x128 panel is electrically a quartet of dinky 240x64 (!) arrays. Four data bits arriving every 960 ns refresh it at about 68 Hz. Each data bit drives a separate quadrant so there are 240 half-speed Dot Clocks on each of 64 rows. Use the -Dot Clock/2 jumper and run the Alternate Frame divider from +Frame Sync+0.5.

The LM215 needs only 15,360 nybbles of data, thus the high-order nybbles can be used for blinking as usual. Because the panel runs at half speed, the dots use only the odd-numbered bytes of the first 30,720 RAM locations. The test pattern for this panel is a little different than the others to cover all four quadrants.

## RELEASE NOTES

The BBS code this month is GRAPH LCD, even though I haven't discussed it. It wiggles the control port, gives you manual control over the 8254 counters, verifies the LCD buffer with pseudorandom data, and writes test patterns into the buffer so you can see your wiring at work. I'll go into more detail next month, but you should be able to puzzle out the essentials on your own.

Be careful with that LCD power supply! Use an ohmmeter to cross-check the LCD pin numbering: the supply voltages and grounds should appear on the filter caps, right? As the woodworkers say, measure twice and cut once... smoking a new panel can really mess up the rest of your day. ☐

*Ed Nisley, as Nisley Micro Engineering, makes small computers do amazing things. He's also a member of the Computer Applications Journal's engineering staff. You may reach him at ed.nisley@circellar.com or 74065.1363@compuserve.com.*

## SOURCES

Last month's column listed sources for LCD panels and other hardware. Check the latest catalogs to find out what's in stock now.

Pico Electronics manufactures ultrareliable DC-DC converters that may come in handy if you're building critical LCD interfaces. Unlike surplus parts, their Series A single- and dual-output supplies come with a pedigree and a price to match, so don't bother them unless you're serious. They're at 453 N. MacQueston Pkwy., Mt. Vernon, NY 10552, (914) 699-5514, fax (914) 699-5565.

Pure Unobtainium has the complete Firmware Development Board schematic and selected parts. Write for a catalog: 13109 Old Creedmoor Rd., Raleigh, NC 27613, phone/fax (919) 676-4525.

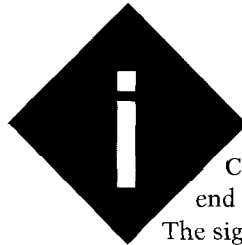
## IRS

413 Very Useful  
414 Moderately Useful  
415 Not Useful

# Warming Up the Digital Thermostat

## FROM THE BENCH

Jeff Bachiochi



visited the Red Cross this weekend to give blood.

The sign-in procedure includes a few tests: your blood pressure, temperature, and extraction of a drop of blood, just to make sure you are in satisfactory condition. I noticed that the usual electronic thermometer was missing. In its place was a sanitary piece of paper wrapped like a band-aid. Once removed from its wrappings, the strip is inserted under the tongue as a thermometer would be. After a minute or so it is removed and small spots [which permanently change color at various temperatures] indicate the temperatures it has been exposed to. Even though they are disposable, there are some positive reasons to use them. They are inexpensive, require no power, are small in size, and are light in weight. They present a good example of today's technology changing the way we measure temperature, even if it is not electronic.

Although a few companies are now announcing thermostatic ICs, Dallas Semiconductor's DS1620 has what is needed to take this idea to the next step. As I discussed last month, the DS1620 offers direct temperature conversion thus

eliminating the conventional analog conversion. It also allows a high and a low temperature setpoint to be adjusted through a clocked serial port. These setpoints govern the hysteresis signature of the output control bit which has a 4-mA source/sink capability. Please see last month's column for more of this chip's particulars.

I chose the primitive 16C56 PIC processor for this project for its low price and availability. It has a single timer and no interrupts, which should prove interesting. The local control and temperature display would fit into the "single page" 16C54 device, however adding a communication interface would easily overflow the 5 12 (12-bit) words available to the '54.

My original concept was to have the thermostat merely display the temperature locally and pass on the data to a centralized controller. The controller would be responsible for calling for heat and controlling the individual zones. This meant if, for some reason, the central controller went down, so did the temperature in the house. To avoid this, I added a bit more sophistication to each thermostat. Now, each can act independently of the central computer, yet still be commanded by the central controller to take advantage of nighttime or unoccupied temperature setbacks.

Jeff adds some smarts to last month's project using Dallas's DS1620 thermostatic device. With a little help from a 16C56 PIC processor, your house will be cool while you sleep and warm while you're up and about.

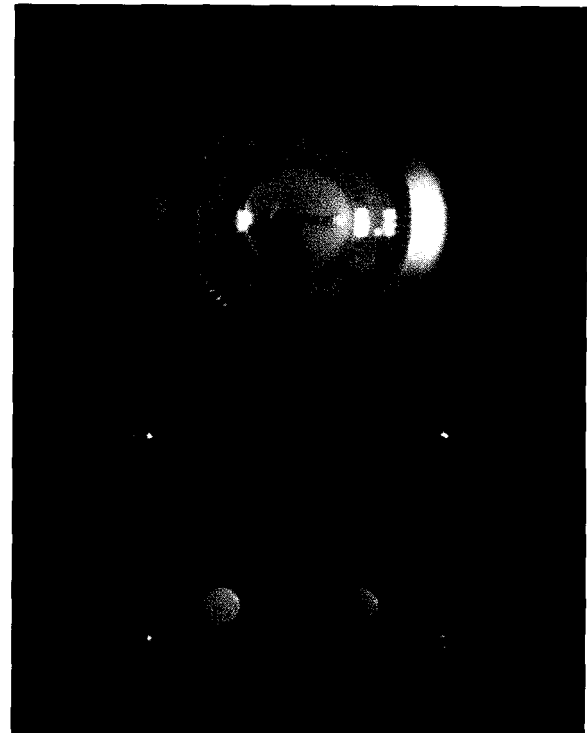


Photo 1—The digital thermostat is mounted below the old one it will replace. The temperature can be read from across the room.

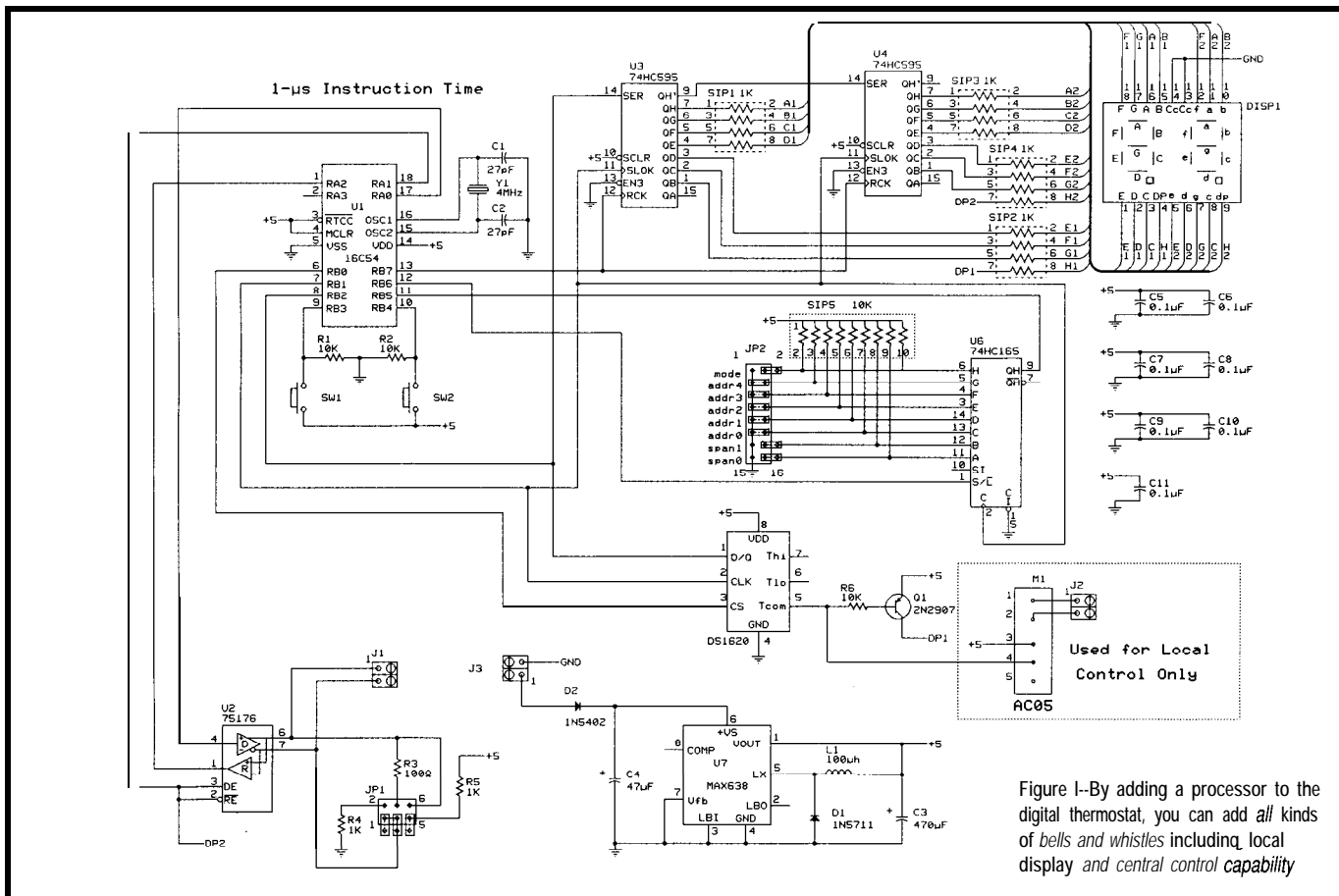


Figure 1--By adding a processor to the digital thermostat, you can add all kinds of bells and whistles including local display and central control capability

## LOCAL I/O

The PIC processor is responsible for five I/O operations. The first is to provide an interface with the thermostatic device, the DS1620. The second is the output to the digital display, and the third is to capture input from the configuration port. The fourth operation is to monitor user input through two push buttons. These push buttons could have been made part of the configuration port, but I didn't want to have to read the port in each time to check the status of the switches. Consequently, they are tied directly to two processor pins. The last operation is communication with the central controller. This is accomplished through a half-duplex RS-485 network (Figure 1).

The configuration port, made from a 74HC165, cannot be tristated, so it cannot share the common data I/O line with the DS1620 and the display port. Configuration data, which consists of temperature span, thermostat address, and display mode settings on JP2, is loaded into the shift register with a \*Load strobe and shifted into

the processor using the common clock line. The span sets the hysteresis setpoints from 1 to 4°C above and below the mean setpoint. The 5-bit thermostat address is from 0 to 31 (that's the maximum number of nodes allowed on a '485 net). The display mode selects between degrees Celsius (the actual conversion) and Fahrenheit (calculated from Celsius).

The display port shares both the common clock and data I/O line with the DS1620. It requires only a single output bit to load the shift register data (16 bits) into the 74HC595's output display registers. The output registers source only about 1 mA to each of the 16 segments. This keeps power dissipation down while still lighting the digits. The left decimal point indicates the thermostat is asking for heat, while the right decimal point shows network activity.

The DS1620 shares both the common clock and data I/O line. It also uses an additional CS (chip select) output from the processor. Since data is passed to and from the DS1620, the

processor must reconfigure its common data I/O line as both an output and an input bit.

The last I/O operation is communication. A 75176 RS-485 driver is used to assure noise-immune communication even at great distances. Only three processor bits-TX, RX, and TX/\*RX enable-are necessary. RX remains enabled until a properly addressed message has been received. A reply is then formulated and TX is enabled only during the response. The net is then free for additional traffic.

## TWO-LEVEL STACK

I do believe the stack implementation is the weakest point of the 16C5x series. Only two levels of calls can be made without some major bending over backwards. Another oddity is that although you can make an LCALL or LJUMP to another page (512-word group) and the page pointers within the status register are correctly set, a RET will pull the complete 11-bit address from the stack. However, in this case the page pointers are *not* set

correctly from the three most-significant bits. Argh!

With only two levels to work with and knowing one of those must be saved for the communication routine, only one call could be made at a time (no nested calls). This makes for some lengthy code.

## °C TO °F

The DS1620 hands off temperature readings in degrees Celsius times two. The "times two" comes about because the least-significant bit of the conversion is a 0.5°C placeholder. If the reading is shifted once to the right, it can be directly interpreted as degrees Celsius. Recall that the Fahrenheit conversion equation is  $°F = (°C \times \frac{9}{5}) + 32$ . Here's how I calculated °F for the limited range of 0–35°C:

- Start the conversion with what is output from the DS1620 [which is actually  $C * 2$ ]
- \*Multiply by 2 (shift left):  $C * 2 * 2$  or  $C * 4$
- \*Divide by 5:  $(C * 2 * 2 / 5)$  or  $C * 4 / 5$

\*Add C:  $(C * 2 * 2 / 5) + C$  or  $(C * 4 / 5) + (C * 5 / 5)$  or  $C * 9 / 5$

•Add 32:  $(C * 2 * 2 / 5) + C + 32$  or  $C * 9 / 5 + 32$

## THE MAIN LOOP

After some initialization code, the thermostat falls into an endless loop. The loop consists of four functions. First, it reads the current temperature from the DS1620. Second, it converts this Celsius value into a Fahrenheit equivalent (if necessary), and also into two decimal digits for display through the 74HC595s. Finally, it checks the push buttons for local requests and if no action is necessary, it simply returns back to the top of the loop,

If at some pass through the main loop either of the push buttons is being pressed, program flow is redirected into the local request loop. The display now flashes with the mean temperature setpoint. If neither button is held down, the display will flash eight times and then return to displaying the actual room temperature. If either the increase or the decrease button is held down, the mean temperature setpoint

(being displayed) will be incremented or decremented appropriately. The new mean temperature setpoint is accepted when the buttons are released. Now the new high and low setpoints will be calculated using the span value (from the configuration port). The DS1620 will be updated only if necessary and the thermostat will return to the main loop.

## NETWORK COMMUNICATION

A network data rate of 9600 bps requires a bit time of 104 μs. Without an interrupt to redirect program flow, a start bit can be easily missed unless the serial input bit is checked frequently. When an interrupt causes redirection, it is normal to wait for 1/2 bit time and then sample the input once each bit time for the rest of the data word. Without an interrupt, how long can we delay before checking (and rechecking) for a start bit?

If the time between checks is limited to a maximum of 1/3 bit time, then delaying 1/3 bit time after the start bit detection will assure the remaining samples are taken between a minimum of 1/3 bit time to a maximum of 2/3 bit time. This falls within a relatively safe sampling region. One-third of a bit time is only about 30 μs. This is an average of about 20 instructions. That means every 20 instructions the start bit must be checked. The main loop and every subroutine it calls must not run more than 20 instructions without checking for a start bit. Needless to say, a lot of time is spent polling.

If a start bit is detected, the communications routine is entered. Serial data bits are collected and are used to build a complete character. The buffer area (part of the 24 bytes of storage available) is shifted and the character is stored at the buffer's end. The buffer is compared for a legal command match and the communication routine is exited if no match is found.

A match consists of a "#" character followed by two ASCII checksum characters, then a space and a "T," which is followed by a two-digit address, a two-digit mean setpoint value, and finally a carriage return. Whew! It looks something like: #49 T0000. If the mean setpoint value is

# Real-Time Multitasking with DOS for Microsoft C, Borland C, Borland/Turbo Pascal

## Develop Real-Time Multitasking Applications under MS-DOS with RTKernel!

RTKernel is a professional, high-performance real-time multitasking kernel. It runs under MS-DOS or in ROM and supports Microsoft C, Borland C, Borland/Turbo Pascal, and Stony Brook Pascal. RTKernel is a library you can link to your application. It lets you run several C functions or Pascal procedures as parallel tasks. RTKernel offers the following advanced features:

- pre-emptive, event/interrupt-driven scheduling
- number of tasks only limited by available RAM
- task-switch time of approx. 6 μsecs (33-MHz 486)
- performance is independent of the number of tasks
- use up to 64 priorities to control your tasks
- priorities changeable at run-time
- time-slicing can be activated
- programmable timer interrupt rate (0.1 to 55 ms)
- high-resolution timer for time measurement (1 μsec)
- activate or suspend tasks out of interrupt handlers
- programmable interrupt priorities
- supports math coprocessor and emulator
- semaphores, mailboxes, and message-passing
- keyboard, hard disk, and floppy disk idle times usable by other tasks
- interrupt handlers for keyboard, COM ports, and network interrupts included with source code
- supports up to 36 COM ports (DigiBoard, Hostess boards)
- full support of NS16550 UART chip
- fast, inter-network communication using Novell's IPX
- runs under MS-DOS 3.0 to 6.0, DR-DOS, LANs, or without operating system
- DOS calls from several tasks without re-entrance problems
- supports resident multi-tasking applications (TSRs)
- runs Windows or DOS Extenders as a task
- supports CodeView and Turbo Debugger
- ROMable
- full source code available
- no run-time royalties
- free technical support by phone or fax

RTKernel-C (MSC 6.0/7.0/8.0, BC++ 1.0/2.0/3.x) \$495 (Source Code: add \$445)

RTKernel-Pascal (TP/BP 5.x/6.0/7.0, SBP 6.x) \$445 (Source Code: add \$375)

For international orders, add \$30 for shipping and handling. Mastercard, Visa, check, bank transfer, or COD accepted.



**On Time**  
MARKETING

Professional Programming Tools

Outside North America, please contact:

On Time Marketing  
Karolinenstrasse 32 • 20357 Hamburg \*GERMANY  
Phone +49 40 43 74 72. Fax +49 40 43 51 96  
CompuServe 100140,633

In North America, please contact:

LEL Computer Systems  
20 Canterbury Court  
Setauket, NY 11733 USA  
Phone (516) 473-8119. Fax (516) 331-0706

"00" then no change is made to the low temperature setpoint or the high temperature setpoint held within the DS1620. Nonetheless, a network response is formulated. The response consists of a "\$" character followed by two ASCII checksum characters, a space followed by a "T," then a two-digit address, a two-digit actual temperature value, a two-digit mean setpoint value, a single ASCII span digit, and carriage return. It looks like: \$7F T0054524.

The central control system can poll each thermostat, optionally setting a new mean temperature setpoint (turning up or down the thermostat). Each thermostat will in turn update the central controller with the present temperature (Celsius ● 2), the present setpoint (Celsius ● 2), and the span (± degrees about the setpoint).

Although each thermostat has the ability to directly control the furnace/zones, you may choose to have the central controller do this (assuming it is installed near the furnace). The central system can accomplish this through mechanical or solid-state relays driven by the controller. This

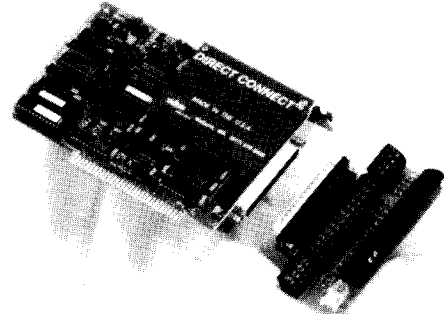
can reduce the number of wires to each thermostat, however it eliminates the ability of each thermostat to call for heat in the event of central controller failure.

The communication routine has priority over the main loop. However, if either of the push buttons is being pressed at the thermostat, communication is ignored while the user makes local temperature adjustments. This assures only one source can make changes to the setpoint at any time.

### SELF HEATING

The circuitry must be as low power as possible to avoid heating up the area where we are trying to measure temperature. The largest radiator of heat would be a simple linear power supply. Since we are starting with 12 volts and only using 5 volts on the board, 7 volts is being wasted across the regulator. To eliminate that kind of heat dissipation, I dug out the MAX638 I used back in my column for issue 38. This is a high-efficiency DC-DC converter with 5-volt regulator. In that article, I used it to make the best use of all the energy

## 16 BIT DATA ACQUISITION?



*Don't Settle For Less Than The Cutting Edge...*

### HIGH PERFORMANCE

Guaranteed 16 bit accuracy  
6 Channel A/D Board **\$1295**

- 16 bit A/D resolution
- 16 bit accuracy
- 50 kHz throughput
- DMA
- 8 lines digital I/O
- 3 channel counter/timer

### LOW COST

16 bit 8/16 Channel  
A/D Board **\$895**

- 16 bit ND resolution
- 15 kHz throughput
- DMA
- 8 lines digital I/O
- 3 channel counter/timer

### OPTIMUM CONVERSION™

DT2601 /571 6 compatible  
ND Board **\$1395**

- 16 bit A/D resolution
- 16 bit accuracy
- DMA, Prog. Gain
- 16 lines digital I/O
- 3 channel counter/timer
- 2 D/A channels

Cut through the specs - each of ADAC's 16 bit boards have been evaluated against every competing model. On noise performance, speed, ease of use, and price, ADAC's leading technology wins every time.

See for yourself - Call for an evaluation board today.

1-800-648-6589

*We've been making data acquisition boards for longer than anyone in the world.*



70 Tower Office Park, Woburn, MA 01801  
FAX (617) 938-6553 TEL (617) 935-6668

Analog & Digital I/O, Industrial PCs, and High Channel Count Systems

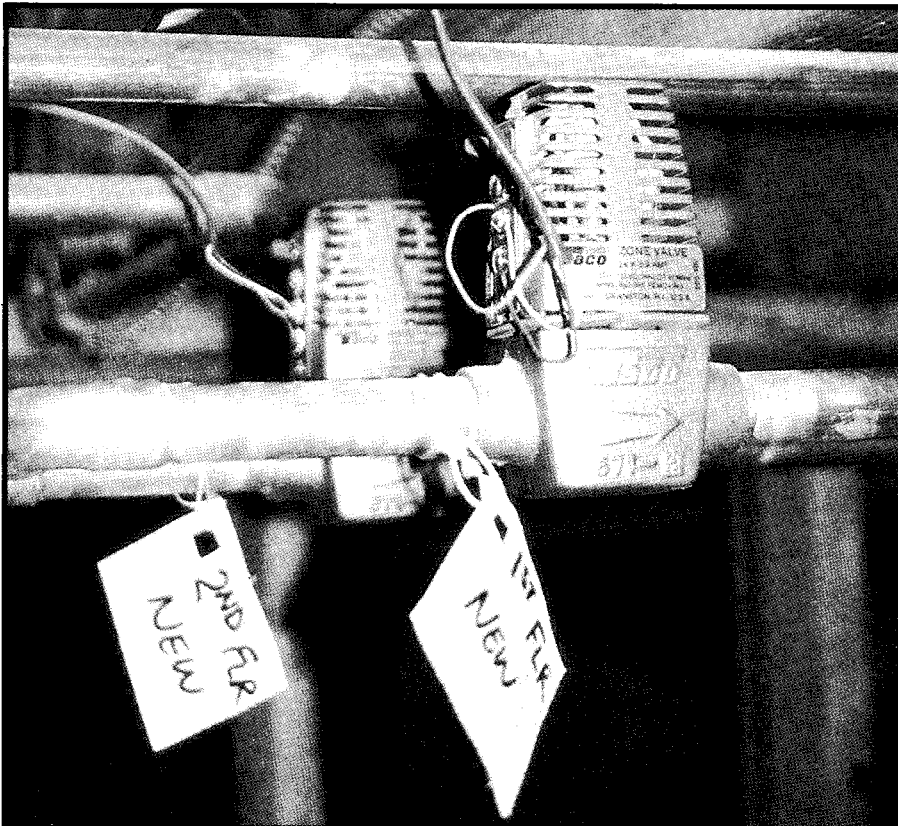


Photo 2—Individual zone valves control the flow of hot wafer to the area of the house requesting heat.

packed inside a 9-volt battery. Here I want to use it for its efficiency as a 5-volt converter/regulator.

The MAX638 is supplied through a series diode to prevent damage from an inadvertent power reversal. All logic devices in the circuit are HC type to further reduce heat. The 100-ohm RS-485 terminator draws more current (50 mA) than the rest of the circuitry put together (although it is supplied only by the active transmitter).

### ELIMINATING THE MERCURY

The familiar glass tube filled with silver fluid may bring back visions of your childhood. Remember when Mom placed the cool stick beneath your tongue to verify your justification for not going to school? Or, seeing the liquid blob roll back and forth to make and break an electrical circuit. These items are slowly finding their way into antique and curiosity shops.

Managing furnace relays and zone valves can require controlling low-voltage AC with operating currents in excess of 1 amp. Most small DIP relays

won't handle the continual abuse of these switching inductive loads. I use a 3-amp optoisolated triac when control from the local thermostat is required. The DS1620's thermostatic output will directly drive the solid-state relay's internal LED. The isolated triac switch can directly drive those control relays and provide isolation between the furnace and central controller.

### SIMPLE (CENTRAL CONTROL) SETBACK SOFTWARE

Setback control can be as simple or complex as you desire. The simplest control consists of turning the thermostat setpoint down every evening after retiring for the night and up prior to rising each morning.

I wrote a BASIC-52 program to communicate with eight thermostats. There are two problems with BASIC-52 for this purpose. The first is that processor bit TO, which enables the RS-485 driver, is not accessible through any BASIC command. A CALL can be used to access a pair of two-instruction assembly language rou-

tines. The first two, CLR TO and RET, place the 75176 (RS-485 transmitter/receiver) into receive mode. The second two, SETB TO and RET, place the 75176 into transmit mode. It turns out not to be such a big deal.

The second problem is much more difficult. The BASIC-52 IN PUT statement must be terminated by a carriage return. If for some reason a reply is not received from a polled thermostat, the system will wait indefinitely. It is true that I could have used a watchdog circuit to whack the reset line if this happened, but the program would only lock up again next time the thermostat did not respond.

The solution I chose was to use Systronix's BCI-51 BASIC compiler. These guys understood the limitation of BASIC-52 when they designed their 8031 integer compiler. The compiled IN PUT routine can be interrupted by either ONEX1 or ONTIME. The second function, ONTIME, will interrupt the IN PUT routine after x amount of time has passed. Since the RS-485 receiver can be always enabled on the RTC52,

## A Notice to our Readers:

The Computer Applications Journal will occasionally provide a listing of subscribers to vendors with offers of substantial interest to our readers.

If you would prefer not to be part of this listing, send the mailing label from the outside cover of the Computer Applications Journal along with your request to:

THE COMPUTER  
APPLICATIONS JOURNAL  
SUBSCRIBER SERVICE DEPARTMENT  
P.O. BOX 7694  
RIVERTON, NJ 08077

### Device Programmers

**Connects to PC parallel printer port**



**48 PIN ZIF**

**\$449<sup>95</sup>**

**EMP-20**

- Easy to use software, on-line help, full screen editor
- **Made in USA**
- 1 & 2 Year Warranty
- Technical Support by phone
- 30 day Money Back Guarantee
- FREE software upgrades available via BBS
- Demo SW via BBS (EM20DEMO.EXE) (PB10DEMO.EXE)
- **E(e)proms 2716 - 8 megabit, 16 bit 27210-27240, 27C400&27C800,**
- **Flash 28F256-28F020, (29C256-29C010 (EMP-20 only))**
- **Micros 8741A, 42A, 42AH, 48, 49, 48H, 49H, 55, 87C51, 87C51FX, 87C751, 752**
- **GAL, PLO from NS, Lattice, AMD-16V8, 20V8, 22V10 (EMP-20 only)**

**SA-20 8 Gang Eprom**



**20 X 4 Line LCD Display**

**\$750<sup>00</sup>**

**40 Pin Keypad**

---

**PB-10 Internal Card for PC**



**2 ft. Cable**

**40 PIN ZIF**

**\$139<sup>95</sup>**

**FOR MORE INFORMATION CALL**

### NEEDHAM S ELECTRONICS, INC.

4539 Orange Grove Ave  
Sacramento, CA 95841  
(Monday-Friday, 8 am-5 pm PST)

C.O.D.

**(916) 924-8037**  
BBS (916) 972-8042  
FAX (916) 972-9960



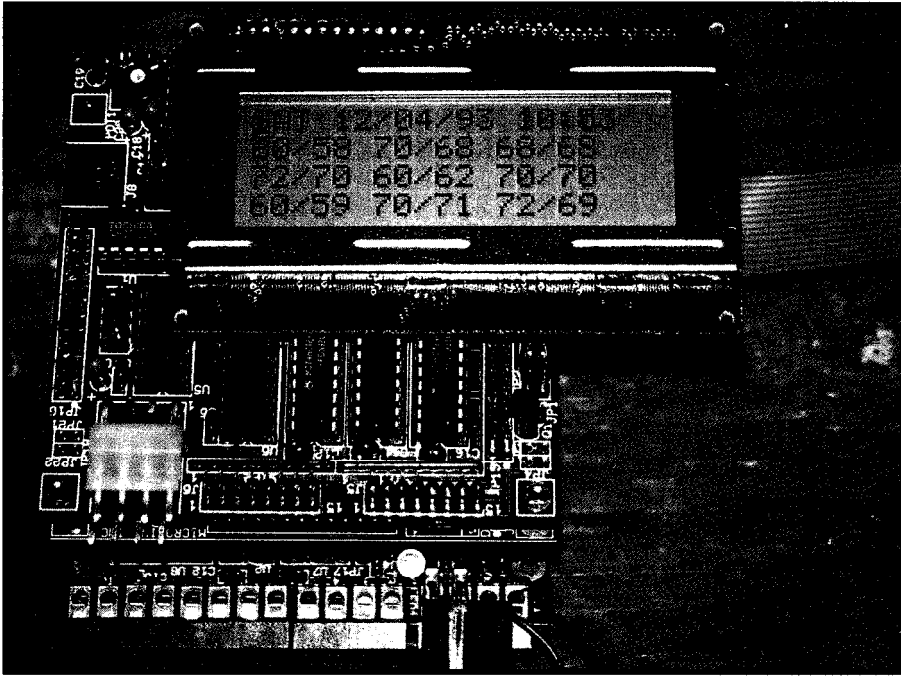


Photo 3—An LCD mounted on the central controller lets you tell at a glance the status of each thermostat

the 0 NTIME interrupt routine can send out a carriage return which will be heard by itself and complete the INPUT statement. If the INPUT string contains only a carriage return, the thermostat is flagged as not responding.

The structure of this program is simple. First, it reads the "time of day" and "day of week" from a clock/calendar. Next it assigns the setpoint times and temperatures from the user's list based on the day of week.

Now it prepares to poll the first thermostat by building a string equal to "#00 T0000." If the present time matches any of the setpoint times, it replaces the last two characters with the new setpoint temperature value. The program adds up all the character values to derive a one-byte checksum (ignoring overflow). Next, it replaces the two "00" checksum digits with two ASCII digits equal to the two's complement of the checksum. Finally, it enables the RS-485 driver, sends out the string, and disables the driver.

The thermostat's return message will indicate the present temperature. I display the present setpoint and the actual temperature (or "no response") if an LCD display is attached. LEDs driven off port 1 indicate where in the polling sequence the program is.

After each thermostat has been polled, the loop returns, the clock/

calendar is read again, and the whole cycle completes.

Remote access to the central control system could be added to give you telephone access to any or all thermostats from the office, airport, or your car. If you're thinking that the HCS II might make a good central controller and would like to see the thermostat supported by XPRESS, please give us your input. Until next month, stay warm. ☺

*Jeff Bachiochi (pronounced "BAH-key-AH-key") is an electrical engineer on the Computer Applications Journal's engineering staff. His background includes product design and manufacturing. He may be reached at jeff.bachiochi@circellar.com.*

## SOFTWARE

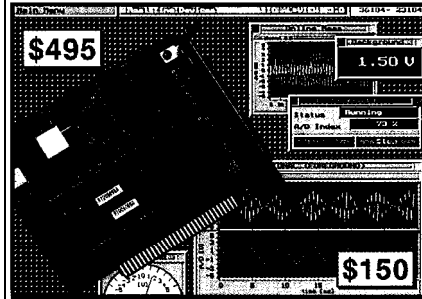
Software for this article is available from the Circuit Cellar BBS and on Software On Disk for this issue. Please see the end of "ConnecTime" in this issue for downloading and ordering information.

## IRS

416 Very Useful  
417 Moderately Useful  
418 Not Useful

## Data Acquisition and Control Without Compromise

**Programmable Scan & Burst**  
RTD's Advanced Industrial Control boards set a new performance standard for general purpose industrial & laboratory applications.



ADA2210 with SIGNAL\*VIEW  
A cost-effective solution

ADA2210 features:

- 125 kHz XT throughput
- 16 S.E. / 8 DIFF 12-bit analog inputs
- $\pm 5, \pm 10, 0$ -+10V selectable input range
- programmable auto channel scan
- programmable burst mode
- software & external triggers, pacer clock
- on-demand DMA transfer
- programmable gain: 1/2/4/8 (1/10/100/1000)
- 3 cascadable 1 B-bit counters
- I/O lines
- two 12-bit analog outputs, selectable range
- rtdLinux Universal TSR DOS driver
- Labtech Notebook driver

### HARDWARE SOLUTIONS

Our AIC family also includes AT I/O boards, +5V only A/D & control boards for sortable PCs, 4-20 mA current loop outputs, and opto-22 compatibility.

### SOFTWARE SOLUTIONS

Select the power and performance of SIGNAL\*VIEW™ or SIGNAL\*MATH™ from our library of application programs for monitoring, data acquisition and analysis, control, DSP and 3D graphics.

For more information on these and other ISA bus and PC/104 products, call, write or fax us today!

**Place your order now and receive SIGNAL\*MATH and SIGNAL\*VIEW for \$195!**



**Real Time Devices, Inc.**

P.O. Box 906  
State College, PA 16804  
(814) 234-8087 ■ Fax: (814) 234-5218

# Op-amp Terminators

## SILICON UPDATE

Tom Cantrell



In our case, the output frequency is now 392 Hz, or 22 Hz higher than in step 2. No explanation can be offered for this difference. . .” This handy little quote can be found in “Design of Op-Amp Circuits,” on page 141, Howard M. Berlin, ISBN 0-672-21537-3. Hey Howard, the explanation is simple—FIRST you stir the entrails, THEN you throw the sticks.

Well-known to be a 1s-and-0s kind of guy, it's not surprising that analog stuff is an occult art to me. The fact that it's also a mystery to the experts is even more interesting.

Sure it would be nice if 1 s-and-0s were enough, but the “real world”—including our own five sensory channels—is undeniably analog. Maybe someday we'll all have “slots” in our forehead, but until then the follies and foibles of analog circuits have to be dealt with.

While I confess to an intellectual curiosity about all this, sometimes (i.e., when the bills are due) I'd rather get something working than explain why it doesn't. Fortunately, as I discovered at a recent Burr-Brown seminar, traditional op-amp suppliers are supplementing their lines with “solution-oriented” parts that take much of the voodoo out of analog design.

### I SEE THE LIGHT

A classic (indeed, it's the same as Figure 4-3 in Berlin's book) op-amp circuit converts the variable current passing through a photodiode to a corresponding voltage using an op-amp. Unfortunately, in the words of

Burr-Brown's experts, it's a solution that's “...often destined for failure...” as shown in Figure 1.

It is interesting to note that you'll find op-amp books, data sheets, seminars, and so forth sprinkled with words like *often*, *might*, *should*, and even *hope*. Welcome to the world of analog design.

According to the experts, the explanation is “...quite simply, the open-loop curve of the amplifier and the noise gain curve of the feedback network have a greater than 20-dB-per-decade difference in their slopes at the intersection.” For those of you who think this explanation isn't “quite simple” enough, the translation is simply “hang a cap on it” (see Figure 2).

Oops, now the bandwidth is all shot to heck. Peering into the crystal ball—yes, I see a faster op-amp in your future. That should do it, eh? So sorry—the new op-amp is faster all right but the signal-to-noise ratio is still poor. Fortune cookie says, “Stay out of debt, use a low-noise FET.” Surely this will solve the problem, right!

But nooo, turns out the input capacitance of the new op-amp is different enough to goof everything up, making the noise worse.

On and on they went, with at least a dozen slides contemplating the thermal noise of resistors, femptoamps per root Hertz of current noise, and other mysteries of the cosmos.

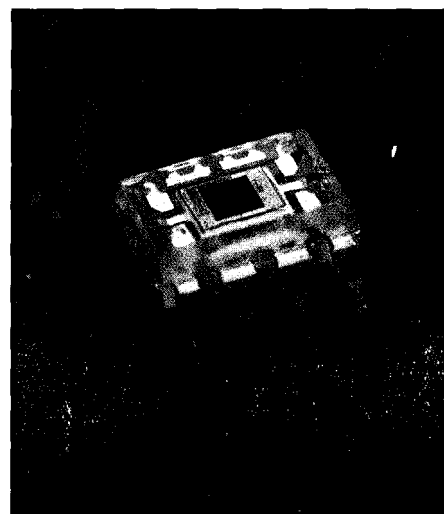
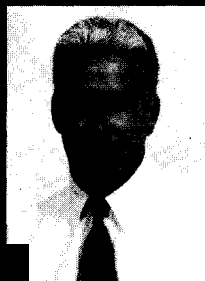


Photo 1—The OPT201 integrates the photodiode, amp, and discretes on a single die in a clear plastic package.



Delving into the occult world of analog

design, our digital hero takes some jabs at this necessary evil of real-world interfacing and checks out some new chips destined to take over where the op-amp left off.

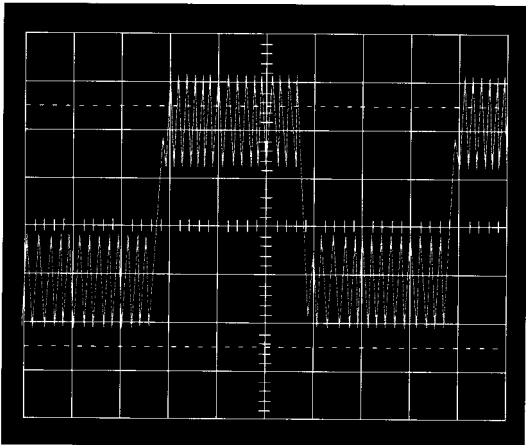
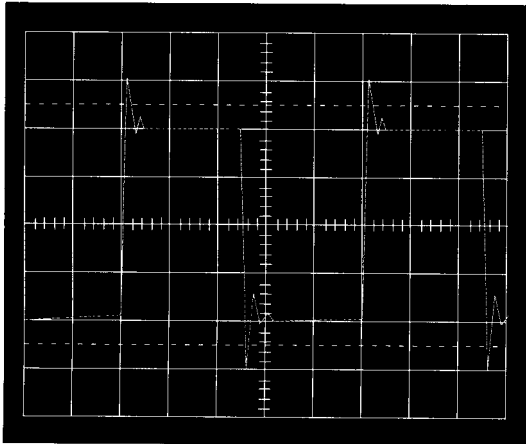
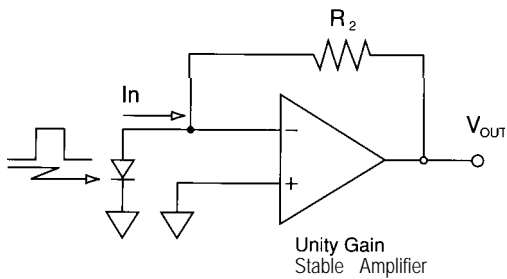


Figure 1-A simple current-to-voltage converter circuit looks good in theory, but in practice fails terribly in response to a pulse input.

I was beginning to fear nothing short of human sacrifice would make the darn thing work. Fortunately, bloodshed can be avoided by using the OPT201 (\$4.35 in 1000s), a device that uniquely integrates the photodiode, amp, and discretes on a single die exploiting the “why didn’t I think of that” concept of using clear plastic for the package (Photo 1 and Figure 3).

The OPT201 is most sensitive to infrared light (770  $\mu\text{m}$  wavelength) and

is specified to achieve 4-kHz bandwidth with a minuscule 30  $\mu\text{V}$  of noise.

### FILTER PRAYERS ANSWERED

Another common op-amp application is to filter signals, passing only frequencies higher than, lower than, or in between the desired cutoffs. Figure 4 (Berlin figure 7-7) shows a universal active (a.k.a., state variable) filter that serves a trinity of applica-

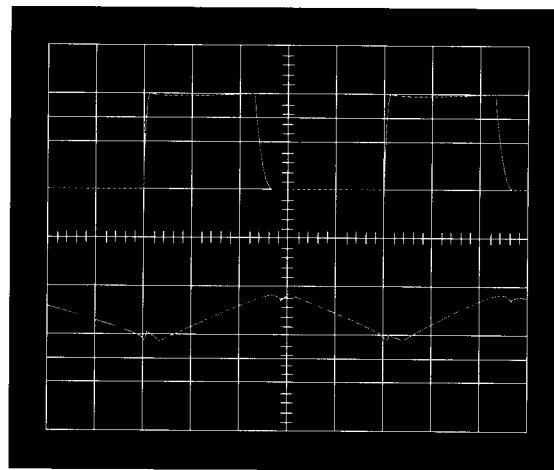
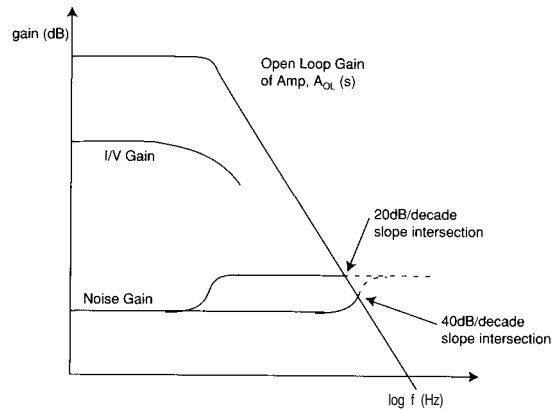
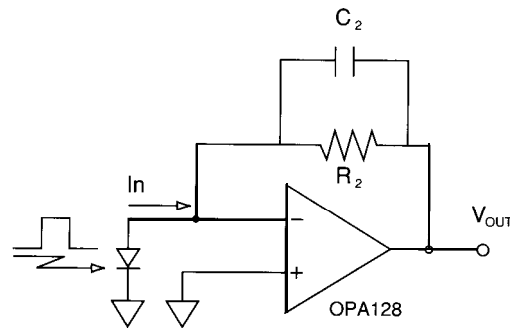


Figure 2-Adding a capacitor to the circuit in Figure 1 cleans up the garbage, but drastically reduces the bandwidth of the circuit

tions, with separate high-pass (VHP), low-pass (VLP), and band-pass (VBP) outputs. In actual practice, a particular setup is usually optimized for either high- and low-pass (and by combining them, notch) or band-pass response.

An immaculate filter would pass everything in the passband and nothing in the stopband while exhibiting no noise, ripple, or ringing to boot. Earthly filters, doomed to sin, make

## FILTER GOSPEL ACCORDING TO...

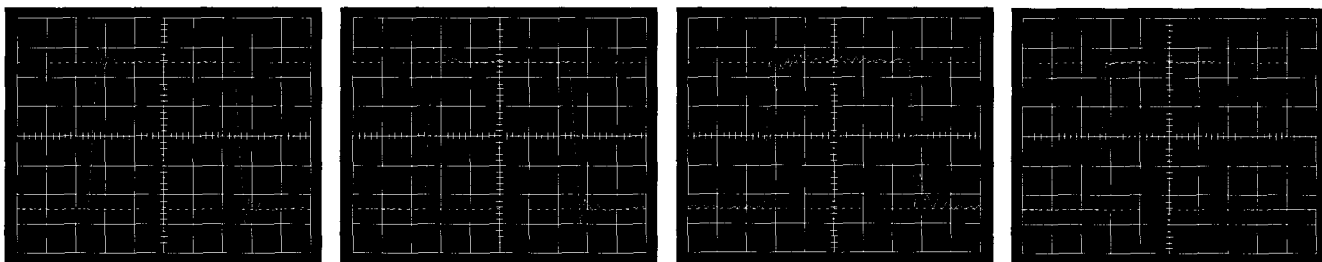
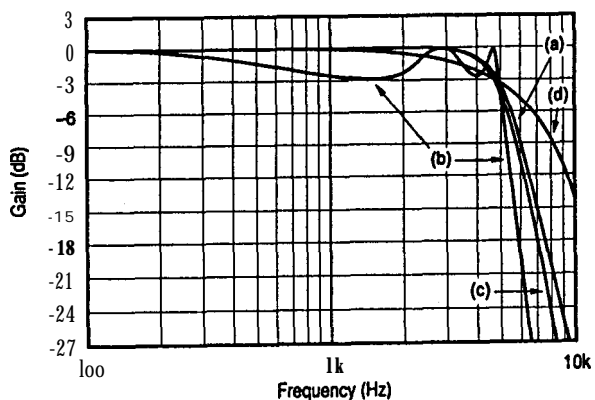
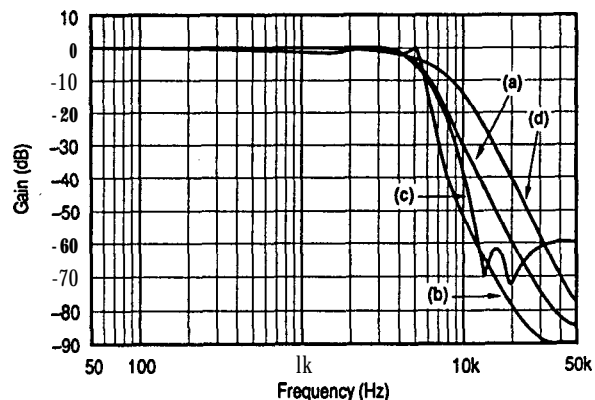
**Butterworth:** Very flat magnitude response (i.e., gain is constant throughout the passband) with fairly steep cutoff (i.e., attenuation in the stopband). Impulse response has moderate overshoot and ringing.

**Chebyshev:** Achieves better cutoff than Butterworth at the expense of some ripple (gain variance) in the passband. Impulse response shows a lot of overshoot and ringing.

**Inverse Chebyshev:** Improves on the Chebyshev by moving the ripple into the stopband and cleaning up the impulse response.

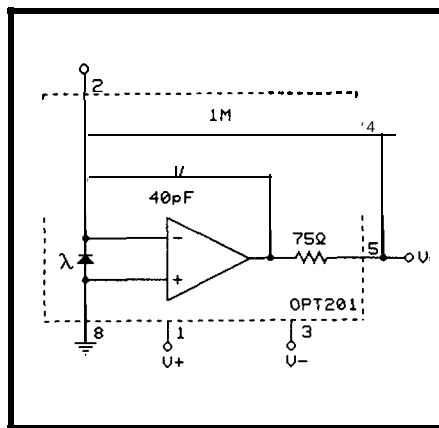
**Bessel:** Achieves the best impulse response with essentially no overshoot or ringing. For a given filter complexity, the magnitude and cutoff response lags Butterworth, but this can be overcome by adding poles. If you're willing to pay the price, a high-order Bessel filter is the way to go.

Both figures to the right show gain versus frequency for fifth-order 5-kHz [a] Butterworth, (b) 3-dB Chebyshev, (c) -60-dB Inverse Chebyshev, and (d) Bessel unity-gain low-pass filters. The top figure shows the overall filter response with the bottom figure shows the transition-band detail. Below are step responses for the same four filter configurations (a-d, left to right).



various tradeoffs between the criteria. So, the first challenge for the uninitiated is to learn the teachings of the various disciples—Butterworth, Chebyshev, Bessel, and so on (see the sidebar)—that testify to a particular filter's fallibility.

Given our previous experience with the simple photodiode amp, it would be miraculous indeed to get the more complicated filter working. Ultimately, a cursed designer might just frantically add more op-amps, resistors, and capacitors in a quick attempt to clean up the mess, ending up with an expensive white noise generator.



**Figure 3**—The OPT201 integrates the photodiode, amp, and discretes on a single die using a clear plastic package.

Analog sinners can be saved with the UAF42 (\$6.40 in 1000s) which combines four op-amps with precision discretes to support a variety (type, number of poles, etc.) of filter designs. Simple filters require only a single UAF42 and two external resistors. More sophisticated (i.e., using more poles) designs are possible using multiple UAF42 chips and perhaps an extra capacitor or two—far fewer discretes (and potential trouble spots) than a traditional op-amp design. Burr-Brown also provides FI LT E R 4 2, a DOS-compatible program which makes designing a particular filter quite easy.

Remember, finally getting a traditional analog circuit working is only half the battle. Keeping it working is another story considering the veritable witch's brew of troubles—sloppy component tolerances, thermal drift, and dubious wiring—that can crop up in production. Your finely tweaked design may go south with the next batch of caps. It's little consolation, but being unemployed will give you plenty of time to explain what went wrong. By contrast, the UAF42 design is blessed with predictable on-chip layout and wiring and tight (laser-trimmed) 0.5% tolerance RC networks.

### GAIN WITHOUT PAIN

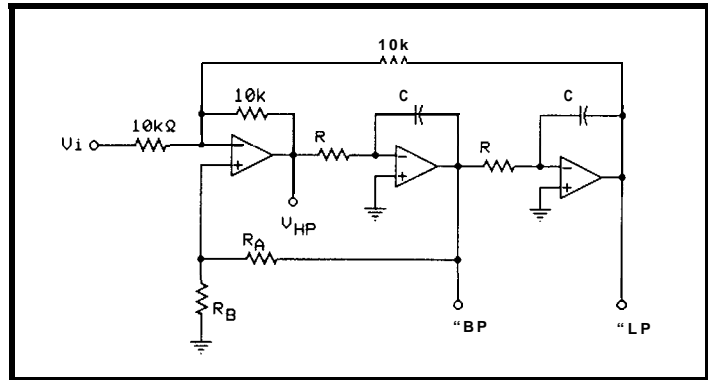
I'm sure the Puritans would rail against such newfangled "easy-to-use" parts. Given we're put on this mortal coil to suffer, it's hard to imagine a more fitting penance than being chained to a balky op-amp design. Besides building character, it makes that inevitable trip to the hereafter start to seem like something to relish, rather than fear.

If we weren't meant to suffer, how come no real-world sensor puts out the 0-5 V a typical A/D converter demands? Interfacing any particular gadget typically requires a special [and expensive] signal conditioner to make the match. Enter the instrumentation amp (IA) which differs from an op-amp in a few key ways.

First, the IA features high common mode rejection (CMR) and thus can dampen noise present on both inputs. For example, connecting a ground-referenced signal to one input and ground to the other can eliminate ground noise of either the periodic (i.e., 60 Hz) or the transient variety. This helps deal with the reality that sensors are usually at the end of noisy cables.

Secondly, an IA includes built-in precision feedback components which, as we've seen, eliminates the potential trouble posed by the external RCs of a regular op-amp design. The previously mentioned CMR also benefits from precision, since even 1% resistance tolerances can shrink it to a scant 46 dB.

Figure 4-A universal active filter provides the designer with separate high-pass, low-pass, and band-pass outputs.



Finally, the inputs of an IA are buffered, presenting a quiet, stable, high-impedance input to the sensor. Note that in an unbuffered op-amp design, the input impedance is low and varies depending on the gain which may negatively impact bandwidth for weak signal sources.

With gain set by a single resistor that has no deleterious effect on the input, one trick is to hang multiple resistors on an analog switch—voilà, you now have an IA with programmable gain.

Better yet, choose a PGA204 or PGA205 (Programmable Gain Amp—\$6.50 in 1000s) which includes the digitally selected gain resistors on chip (Figure 5). The '204 features gain selections of 1, 10, 100, and 1000 while the '205 offers gains of 1, 2, 4, and 8. You can cascade multiple units for more range—for example, a '204 and a '205 in series expands gain selections to 1, 2, 4, 8, 10, 20, 40, 80, 100, 200, 400, 800, 1000, 2000, 4000, and 8000.

A plus for the '204/'205 chips is that the output limit is dictated by the

chip's power supply, so you don't need to worry about a software error (choosing too much gain) feeding 8000 V to your A/D converter. Furthermore, the inputs are protected up to 40 V, a feature that not only adds robustness but can eliminate the need for extra input limit circuits.

*Tom Cantrell has been an engineer in Silicon Valley for more than ten years working on chip, board and systems design and marketing. He can be reached at (510) 657-0264 or by fax at (510) 657-5441.*

### CONTACT

Burr-Brown Corporation  
P.O.B. 11400  
Tucson, AZ 85734  
(602) 746-1111

### IRS

419 Very Useful  
420 Moderately Useful  
421 Not Useful

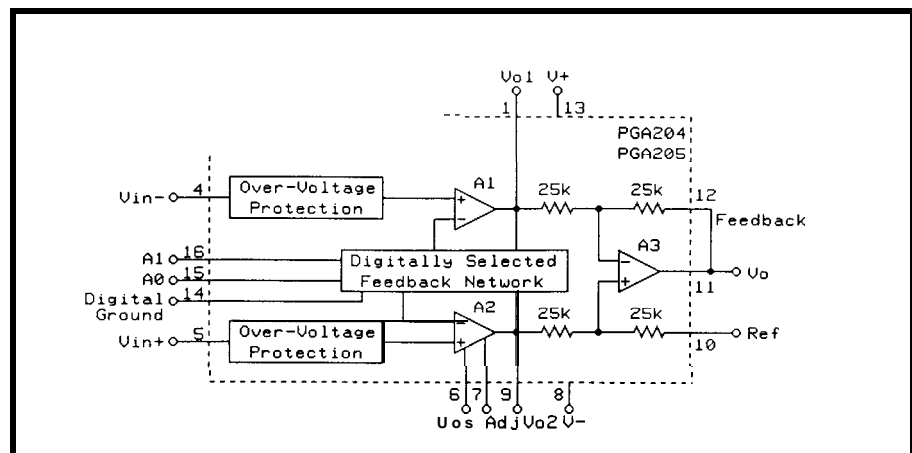
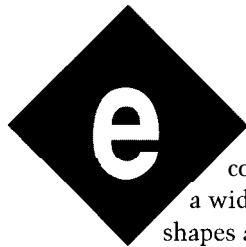


Figure 5—The PGA204/205 provide the ultimate in flexibility with their digitally selectable gains, and also offer low noise and high common mode rejection

# Data Logging and Collection Devices

## EMBEDDED TECHNIQUES

John Dybowski



Embedded controllers come in a wide variety of shapes and sizes. This is only natural since the applications they are pressed to serve are equally diverse. Among the many applications for embedded controllers are appliance control, medical instrumentation, down-hole monitoring, and solar-powered weather surveillance systems.

Looking at embedded controllers from another perspective, the differentiating features are often perceived to revolve around the controller's underlying technology and architecture rather than its eventual usage. These days, 32-bit controllers are becoming commonplace in applications that were simply not possible before their arrival.

Of course, this is not to say that their smaller 16-bit, 8-bit, and 4-bit siblings are faced with extinction. Designers of embedded systems are a practical lot, and many, if not most, embedded systems can be served perfectly well using 8-bit controllers. After all, if all you need is a 2N2222 transistor, then that's what you will use. Similarly, you may find that an 8051 has all the horsepower needed for a given application. Why fight it?

### I'VE FALLEN AND I CAN'T GET UP

The balance of power has actually come full circle if you consider that parts like the 8051 still prevail in many new embedded designs. Evidently, those who proclaimed the death of 8-bit CPUs were, shall we say, a bit premature. What's surprising is not so much the fact that the 8051 is 15 years old, but that new derivatives

based on this basic architecture are still being introduced on a regular basis. Many of these newer 8051 derivatives contain additional built-in features. As a result, some of these parts can get pretty expensive. The important thing here is that this architecture is familiar to many engineers and the tools are in place to expeditiously turn designs around. Maybe you started using the 8051 because you needed a \$2 controller, but the situation is a bit different now. Evidently though, once you've taken the plunge, it's hard to go back.


It should be apparent that some problems require the aid of a high-tech, 16- or 32-bit powerhouse to bring them into check. It is equally clear that many applications can be put to rest by applying relatively low-tech solutions. The median is, of course, the domain of mid-tech computing devices, which is the place between the perils of the leading edge and the trailing mundane. This is a comforting thought for many engineers since this middle ground is a relatively safe place. It is founded on familiar and stable technology, yet still possesses the potency to manage many challenging design problems.

I should back up now and touch on some common functions that are useful in a variety of control tasks. These function blocks form the basis of many general-purpose embedded computers. Also, it would be wise to temper this discussion with the knowledge of what constitutes modern design practices and how these impact the fundamental embedded computer requirements.

The basic requirements of a general-purpose embedded computer would include the following items:

- \*Real-time clock/calendar
- \*Digital I/O
- \*Analog I/O
- \*Nonvolatile RAM
- RS-232 or RS-485 serial ports
- E2PROM for parameter and setup information
- Basic expansion capabilities

In addition, the following elements could be used in certain applications:



So what comes between a high-tech Cray and a low-tech transistor? Follow along as John starts a series describing a general-purpose embedded control system with lots of potential.

- A display and keyboard
- Some sort of mass storage system
- \*A watchdog timer
- \*The capacity for battery operation

And for those applications that relied on extended battery life:

- \*All-CMOS design
- Power management
- \*Battery management

## THE NATIVE 8031

In spite of all the peculiarities of the 8031 architecture, I've always been impressed with its bit manipulation capabilities. This appreciation came after my initial horror of its twisted little architecture faded. In contrast to the rigors of accessing the 8051's external data space, manipulating the bit-addressable port pins is relatively straightforward. Sadly, most implementations that use the 8031 in expanded mode sacrifice many of the I/O pins to their alternate functions as the data bus, address bus, and other memory control functions. As a result, you frequently end up adding memory-mapped I/O, which leads you back to the rigors of accessing the 8031's external address space. I'm sure you appreciate the nature of the problem.

Those of you who have followed my columns may recall that, on occasion, I've managed to circumnavigate this problem using controllers such as the DS5000 and DS2250 from Dallas Semiconductor. Essentially possessing the fundamental elements of some of the single-board computers on the market, these controllers provide an ideal foundation on which to build a compact and flexible embedded computer. Some complain these parts are overpriced, but I'm

amazed that these same people would not hesitate spending an equal amount of money on some lame single-board computer that offers only a fraction of the functionality of these highly integrated controllers. It's a curious state of affairs, and I suppose it all boils down to a matter of perception. Some folks take an 8031, an address latch, and an EPROM and call it high tech. Most of us know better.

Now that I've stated why I think these Dallas controllers have the characteristics on which to base a general-purpose embedded computer system, let me briefly recap what these parts bring to the party. First off, the

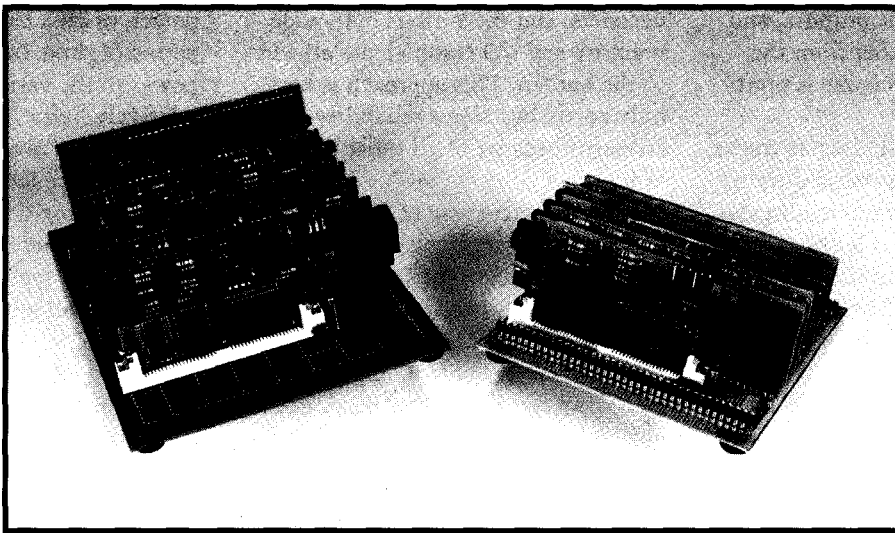


Photo 1-Although the chief purpose of this embedded computer is for data collection and data logging, experienced engineers will find this system useful for other control system applications.

DS5000 and DS2250 are both based on a Dallas DS5000FP processor core which, for all intents and purposes, looks a lot like a souped up 8031. The DS5000 encapsulates this core processor along with up to 32K of RAM, an optional real-time clock, and a lithium backup power source (rated at ten years) within a 40-pin DIP package that maintains pin and function compatibility with a standard 8031. The DS2250 provides the same functionality along with a maximum of 64K RAM on a less-expensive 40-pin SIM affair. (Check out my "ONDI" [issues 16 & 17] and "The Elements of a Data Logger" [issue #27] articles for additional insight into the inner workings of the DS5000 and DS2250.)

Other than the differences in packaging and the maximum amount

of available memory supported, both parts are functionally equivalent. Since the built-in RAM is accessed via internal data, address, and control buses, this leaves all of the on-chip ports available for use as general I/O. Finally, we have a platform that allows us to have memory and I/O at the same time! Besides all this, the lithium-based RAM can be partitioned as either program memory or data memory, which allows using this memory in the most efficient manner for the particular application. This partitioning can be performed either during initial program loading or dynamically under firmware control in reasonably small increments.

Additional features such as a built-in watchdog and power-on reset, power-fail interrupt, and additional specialized SFRs provide functionality that is quite useful in many embedded computer applications. An integral bootstrap loader allows serial or parallel program loading,

memory partitioning, program verification, and setting various chip configuration options from your PC. This capability essentially eliminates the need for EPROM emulators that are frequently used during initial program development. It turns out this feature can be put to good use when developing programs that will run on parts that don't support any external program memory at all. For example, when I was toying with an 8751 project a while back, rather than burn 8751s, I just used a DS5000 and downloaded the code until I got it right. Once I knew it worked, I then burned my 8751-once.

## WORK THOSE BITS

Using the DS2250 with all of those easy-to-use I/O bits lets you

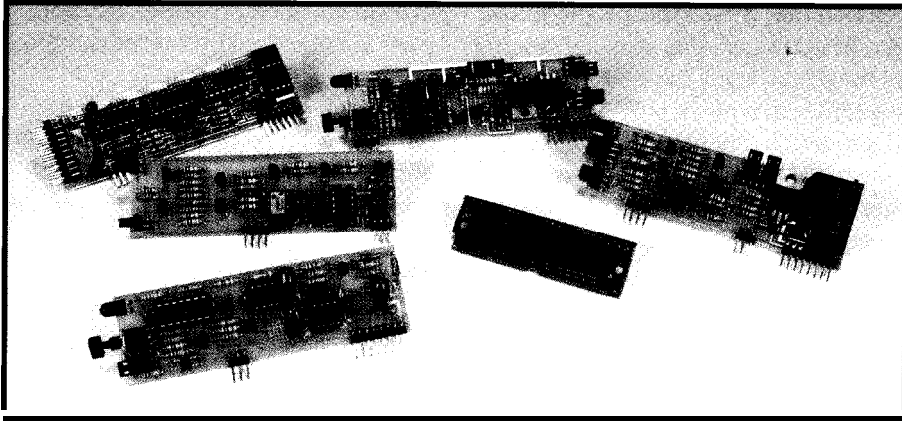


Photo 2-System elements include the processor board, power I/O, power manager, PC core, BIONet, battery manager, user I/O, and BIONet satellite.

proceed in two different ways. If what you're doing is developing a controller to satisfy some specific purpose, and you're putting it together from the ground up, then your course is pretty well charted.

On the other hand, if what you're doing is developing a general-purpose embedded computer, then it's impossible to anticipate what burdens may ultimately be placed on the instrument. Naturally, you have to provide some basic feature set while leaving room for future expansion. At the same time, it's advisable to limit your focus and set some reasonable limits on the device's eventual capabilities. If you try too hard to design an embedded computer that is all things to all people, you may find yourself the hapless creator of something that is ultimately nothing to everybody. You've got to walk the line.

Assuming you've set your sights properly, it makes sense to subscribe to the old engineering adage of max-for-min. The key is to not paint yourself into a corner and also to provide a reasonable amount of flexibility without getting carried away. Developers of embedded computers have traditionally countered this quandary by providing some sort of bus architecture in order to furnish an avenue for handling options as needs arise.

This is a reasonable idea, but it is not entirely without problems. By its very nature, such an architecture usually involves numerous connections to carry the processor data and address bus as well as the required

control signals. To support signaling across multiple boards, buffers, decoders, and the various byte-wide memory and I/O components all add to the burden. This approach is not only costly but raises reliability concerns because of all those connectors.

An alternate way of dealing with the expandability issue is to use the on-chip I/O bits of the DS2250 as a sort of uncommitted bus. Since these bidirectional pins operate completely under firmware control, ultimate flexibility can be realized. Each pin can be dynamically defined as input, output, or bidirectional. The timing of each pin can also be controlled to suit the individual peripheral function it is assigned to control. Since these functions operate under firmware control, a modest level of throughput is implied. But remember that the DS2250 controller is self contained and runs at the full bus bandwidth. If this all sounds a bit familiar, then you've been following my recent columns where I've demonstrated how to control a multitude of functions serially with a basic 8031. Admittedly I've approached this subject piecemeal and somewhat unevenly thus far, but now I'm ready to come clean and show you how to put it all together.

### A COMPACT EMBEDDED DATA COLLECTION COMPUTER

I'd like to take a little time to describe the basic system configuration for a general-purpose data collection computer so we can operate from a common baseline.

This embedded computer is centered around a DS2250 controller with 64K of battery-backed memory and running at 11.0592 MHz. The computer is based on a carrier card (something of a backplane) that holds the DS2250 and all of the feature cards. The feature cards sit vertically atop the carrier and allows the system to have a very small footprint. A 3% conductor bus is defined to consist of 24 general-purpose I/O bits, a couple of dedicated control bits, a number of power connections, and several card-to-card control signals.

The system is designed for low-power operation. The basic configuration is capable of running from line power, or off a battery for an extended period of time. In order to supply power to the various feature cards, a relatively hefty 7.2-volt, 750-mAH NiCd battery is specified. This battery should suffice for most applications. But depending on which features you want to use and how you want to use them, many applications could work perfectly well with a much smaller battery.

Although intended for general use, this embedded computer is ideally suited for data collection and data logging. Photo 1 shows a typical system configuration. Also presented in this photograph is my initial handwired prototype that served as the test bed for my development. In Photo 2, the system is dismembered so you can see the cards that comprise the system. As I briefly describe each card, see if you can pick it out of the lineup.

The Power I/O card provides a line-powered RS-232 port, a CMOS RS-485 port, an I<sup>2</sup>C port, and a line power connector. Also on this card is a +10-V regulator that preregulates the line power to a level that the micropower switch-mode regulators (that reside on the other cards) find palatable. Finally, a couple of jumpers are provided to support the system bootstrap loader. These jumpers allow you to control bootstrap mode. You can set the jumpers so that the bootstrap load can be invoked via the Loader Utility from the host PC, or you can completely inhibit this capability, or you can invoke it



manually. The LED on this card signifies that bootstrap mode is in effect when it is lit.

The Power Manager contains two voltage regulators for the system. One of them is a micropower pass regulator running at +3.5 V that sources backup power for nonvolatile peripherals, and the second one is a micropower

switch-mode buck regulator for supplying the +5 V used to power the system's logic. If you were wondering about the +10-V preregulator, it is there because the buck regulator used in this design tends to self-destruct if hit with more than +12 V.

This card also contains control logic that serves to control the +5-V logic regulator. This logic turns on the system power in response to the application of line power, a pulse on the manual push-button switch, or via a signal generated by one of the other feature cards. Shutdown is accomplished by the system processor via a port pin. The LED indicates the system is active and is on when the main +5 V is present. An output is provided to the system processor that indicates that the system is operating under battery power. Another output signals when the battery is failing and warns that the main +5-V switcher is about to lose regulation.

The I<sup>2</sup>C core contains all of the local I<sup>2</sup>C peripheral functions and provides the majority of the system I/O. Contained here are 8 bits of bidirectional digital I/O, 8 channels of 8-bit A/D conversion, and 2 channels of 8-bit D/A conversion. The analog section operates at levels of 0 to 2.5 V. All of the digital and analog I/O signals are brought out to a 20-pin, double-row header. Channel 0 of the ADC is configured to monitor the battery and



Photo 3—The complete system includes processor board, interfaces, displays, keypad, sensors, and battery power management.

allows the processor to monitor the battery's discharge progress.

Also riding the I<sup>2</sup>C bus are a 512-byte E<sup>2</sup>PROM and a real-time clock. You may wonder why the E<sup>2</sup>PROM would be included in this configuration since the DS2250 already has a significant amount of nonvolatile RAM. The answer to this question is that everything in the DS2250's memory is lost when you use the bootstrap loader to download programs to the system. In this case, it's beneficial to have a separate holding area for configuration, setup, and identification information.

The RTC is not only the time-keeper for the system, but also generates the enable signal that is ultimately routed to the power control logic and is used to bring the system to life. This RTC alarm function is very flexible and the signal can be programmed to fire off on a specific date and time, or it can be configured to operate as an interval timer with increments anywhere from a fraction of a second to a number of days. This alarm output is buffered and presented to the system processor and can be polled or used as an interrupt source. Finally, the RTC contains 256 bytes of nonvolatile RAM that can be used for additional miscellaneous storage.

Extended digital sense and control points are handled by the BIONet card. On occasion, there is a need to

monitor and control digital points that are dispersed over a wide area.

Usually this must be handled by bringing numerous cable bundles back to the central controller. This card, along with its associated satellite peripherals, alleviates this problem. Containing a micropower +15-V switch-mode boost regulator

and related interface circuitry, up to 60 inputs and 60 outputs can be serviced over a single twisted-pair cable. The interface allows driving the data and power over these two leads. This configuration is optimized for low-power operation and allows moderate line lengths from 250 to 500 feet. The on-card LED indicates when the network is active.

Since the system is fully capable of operating from battery power for extended periods, it makes sense to show some kindness to the battery. The battery manager is configured for fast charging a NiCd battery using constant current. Primary charge termination is achieved using the negative delta voltage method and requires no temperature sensors. For safety, charge is unconditionally terminated following a maximum time period should the primary cut-off mechanism fail.

For battery conditioning, and to ward off the feared memory effect, discharge-before-charge can be initiated by activating a manual push-button switch. The LED indicates various charge functions via different sequences of on/off patterns.

User I/O is important, and this embedded computer allows some flexibility in this regard. The LCD/keyboard interface is an PC-based component that can connect to the I<sup>2</sup>C port of the Power I/O card if you want

to operate this section remote to the controller. The interface card consumes two I<sup>2</sup>C addresses when the keyboard is included and only one if just the LCD is used. In any case, a number of free addresses are available so this is not a problem. This card is the same size as a typical 20x4 LCD and attaches discreetly beneath the display panel. A 4-bit digital port is provided that allows connecting a beeper and other extra devices such as LEDs. The I<sup>2</sup>C bus is carried through to a second connector to allow daisy chaining additional devices if necessary. The complete system shown in Photo 3 uses this feature to handle a fully independent secondary LCD. Even if your end product does not need such capabilities, it is quite a useful feature to have during system debugging.

Now that I've described the cards that comprise the basic system, let me summarize each card individually. Just so we can keep score, I'll also note the number of I/O pins each card consumes.

### POWER I/O:

- @Line-powered RS-232 port
- \*CMOS RS-485 port
- \*Host control of bootstrap loader (with jumper defeat)
- \*Manual control of bootstrap loader
- \*I<sup>2</sup>C port for LCDs, keyboards, and other peripherals
- +10-volt preregulator for micropower switchers
- I/O pins used: 3 for SIO (I<sup>2</sup>C port uses same 2 pins as I<sup>2</sup>C core)

### POWER MANAGER:

- +5volt micropower switch-mode buck regulator
- +3.5-volt micropower continuous-mode regulator for backup power
- \*Battery/line signal
- \*Battery fail signal
- \*Power control logic turn-on via:
  - Programmable RTC/timer
  - Manual push button
  - Attachment of line power
- \*Power control logic shutdown under processor control
- I/O pins used: 3

### I<sup>2</sup>C CORE:

- 8 channels of 8-bit ADC with an input range of 0-2.5 V
- \*Battery monitoring via ADC channel 0
- 2 channels of 8-bit DAC with an output range of 0-2.5 V
- \*Real-time clock/calendar
- 5 12 byte E<sup>2</sup>PROM
- 256 bytes of RAM
- 8 bidirectional digital I/O lines
- \*Programmable interrupt source
- \*Programmable RTC/timer wakeup stimulus
- I/O pins used: 2

### BIONET:

- Processor-controlled micropower switch-mode boost regulator +15 v
- BIONet port for up to 60 remote I/O points over twisted pair
- I/O pins used: 4

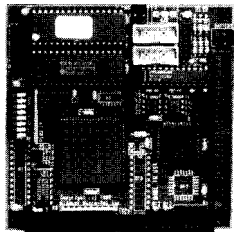
### BATTERY MANAGER:

- \*Constant-current source charger
- Fallback to trickle charge
- \*Charge termination via negative

## We're Small, We're Powerful, And We're Cheaper.

### MMT-188 EB

- 2 serial I/O ports
- 3 programmable parallel I/O ports
- 1 Meg RAM/ROM capable
- powerfail detect interrupt and reset
- counter-timers
- watch dog timer
- expansion connector



Only \$191<sup>00</sup> (Qty 100)

ALSO AVAILABLE: MMT-Z180, MMT-196, MMT-HC11, MMT-EXP

In fact, you'll get the best product for about half the price. If you're interested in getting the most out of your project, put the most into it. For the least amount of money.

Call us today for complete data sheets, CPU options, prices and availability.

### Custom Work

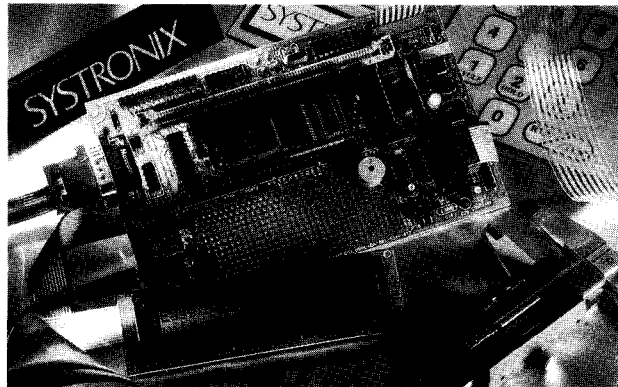
Welcome. Call or fax for complete data sheets

2308 East Sixth Street  
Brookings, SD 57006  
Phone (605) 697-8521  
Fax (605) 697-8109



WE'RE SMALL BUT WE'RE POWERFUL.

## NEW! UNIVERSAL DALLAS DEVELOPMENT SYSTEM from \$199!



- It's a complete single board computer!
- One board accommodates any 40 DIP DS5000, 40 SIMM DS2250, 40 SIMM DS2252, or 72 SIMM DS2251, 8051 superset processor! Snap one out, snap another in.
- Programs via PC serial port. Program lock & encrypt.
- LCD interface, keypad decoder, FE.232 serial port, 8-bit ADC, four 300 mA 12V relay driver outputs.
- Power with 5VDC regulated or 6-13 VDC unregulated
- Large prototyping area, processor pins routed to headers
- Optional enclosures, keypads, LCDs, everything you need
- BCI51 Pro BASIC Compiler w/50+ Dallas keywords \$399

**SYSTRONIX**® TEL: 801.534.1017 FAX: 801 534. 1019  
555 South 300 East, Salt Lake City UT, USA84111

delta voltage method and/or maximum time

- \*Discharge-before-charge via manual push button
- I/O pins used: 0 (stand-alone)

These five cards form the fundamental peripheral set for the system, with each card being 1" x 4" in size. The system carrier contains the DS2250 and nine peripheral slots. This card measures 4" x 5". The configuration 1 described leaves four free slots open for future expansion. The basic system accepts the following standard remote peripheral components:

#### USER I/O:

- I2C addressable 4x20 LCD
- I2C addressable 4x4 keypad
- \*Multiple user I/O cards can be accommodated by the system

#### BIONET SATELLITE:

- 2 input bits
- 2 output bits
- Up to 30 addressable satellites
- Power and data over twisted-pair

network that can be up to 300 feet long.

### THE EVOLVING CONTROLLER

The embedded computer I've presented is designed to serve experienced engineers as a vehicle for their specialized data collection and control applications. In subsequent columns, I will describe in detail both the electronics and firmware associated with each peripheral function. I'm eager to move along and get on with developing some real applications in order to demonstrate how this embedded computer can be put to use in various data collection, analysis, and control tasks.

In order to make short work of this, I'll be doing a lot of the coding in C, leaving the assembler for the real low-level stuff. I did mention something about modern design practices, didn't I? With the extra elbow room of the DS2250's 64K of memory, this shouldn't be a problem at all. The system's bootstrap loader will further serve to expedite the whole operation.

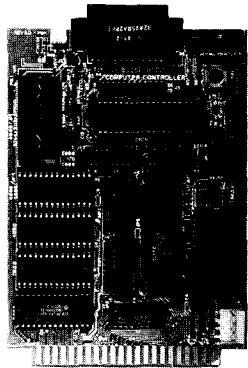
Of course, this is not a static system, and it will continue to grow and evolve. After all, that's the whole idea. Presently, a memory expansion card is under development that will offer 512K of battery-backed SRAM for more ambitious data collection tasks. To go hand-in-hand with more ambitious undertakings, a 16-channel analog input card with 0-4.096-V input span and 12 bits of resolution is in the planning stages. What's next? A smaller battery power option and perhaps solar power—the embedded computer goes green. □

*John Dybowski is an engineer involved in the design and manufacture of hardware and software for industrial data collection and communications equipment. He may be reached at john.dybowski@circellar.com.*

## IRS

- 422 Very Useful
- 423 Moderately Useful
- 424 Not Useful

## BCC52 BASIC-52 COMPUTER/CONTROLLER



The BCC52 controller continues to be Micromint's best selling single-board computer. Its cost-effective architecture needs only a power supply and terminal to become a complete development system or single-board solution in an end-use system. The BCC52 is programmable in BASIC-52, (a fast, full floating point interpreted BASIC), or assembly language.

The BCC52 contains five RAM/ROM sockets, an "intelligent" 27641128 EPROM programmer, three 8-bit parallel ports, an auto-baud rate detect serial console port, a serial printer port, and much more.

#### PROCESSOR

- 80C52 B-bit CMOS processor w/BASIC-52
- Three 16-bit counter/timers
- Six interrupts
- Much more!

#### MEMORY

- 48K RAM/ROM, expandable
- Five on-board memory sockets
- Either 8K or 16K EPROM

#### Input/Output

- Console RS232—autobaud detect
- Line printer RS-232
- Three 8-bit parallel ports
- EXPANDABLE!
- Compatible with 12 BCC expansion boards

**To Order Call 1-800-635-3355**  
Tel: (203) 871-6170  
Fax: (203) 872-2204

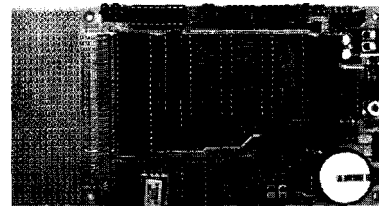
BCC52	Controller board with BASIC-52 and 8K RAM	\$189.00 Single Qty.
BCC52C	Low-power CMOS version of the BCC52	\$199.00
BCC52I	-40°C to +85°C industrial temperature version	\$294.00
BCC52CX	Low-power CMOS, expanded BCC52 w/32K RAM	\$259.00

CALL FOR OEM PRICING

**MICROMINT, INC.** 4 Park Street, Vernon, CT 06066  
in Europe: (44) 0285-658122 • in Canada: (514) 336-9426 • in Australia: (02) 888-6401  
Distributor Inquiries Welcome!

#119

## 8051 EMBEDDED CONTROLLERS With Lots of Extras!



We have a full line of low cost 80C32 embedded controllers and software tools which are ideal for

#### Features include:

- Low power CMOS design
- Up to 60K of code space and up to 60K of data space
- 5 to 15 volt operation
- Small form factor (3.5" x 6.5") with prototyping holes

- Start at \$100

#### Available Options:

- Multifunction Board adds A/D, 24 I/O lines and timer
- BASIC-52 or Monitor/Debugger in EPROM, Compiler \$100 or BASIC Compiler for \$300

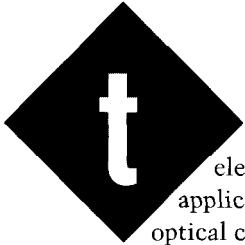
#### Iota Systems, Inc.

POB 898 Incline Village, NV 89452  
PH: 702-831-6302 FAX: 702-831-4629

#135

# PATENT TALK

by Russ Reiss



There are many uses for light in electronic systems. Some typical applications include coding of information, optical communication, signaling, or used simply as an output annunciator. This month I have collected eight very recent patents from the 1993 Patent Database which all deal with applications involving light in electronics.

The first patent in Abstract 1 is perhaps the simplest. It uses light reflected from bowling pins to determine which are standing and which have fallen. A lens focuses the light onto a linear light sensor array; and (of course) a microprocessor is called into play to extract the information from this sensor and to communicate the results over a serial link. Mechanical alignment and proper lighting would seem crucial to the success of this technique, but it is a novel and potentially robust and low-cost approach.

The next two patents use the color of a light source, or of a lighted object, to encode and represent information. Abstract 2 is a unique and sophisticated imaging system which makes use of a most simplistic light source. There is often the need to rapidly measure audience response in a binary manner [e.g., yes/no]. This patent uses colored reflectors (red on one side and green on the other) which are held up by the audience to indicate their preference. A video camera scans the scene and a computer picks out the red and green "dots" from the background and counts them. This technique avoids having to wire the seats or provide selection buttons at each site. It is very simple to use, yet provides rapid response.

I can envision a number of enhancements. Reflectors like these found on automobiles and road signs would

return a stronger signal. Using colored flood lights matched to the objects' emissivity would enhance the image of the objects. Increasing the measurement time or doing multiple counts might account for movement or misalignment. It would also seem possible to use more than just two colors where a more complex vote is required, although this would require more complex (or multiple) signaling objects. While the accuracy may not be 100%, the system is simple and fast, and probably quite adequate in many situations.

In contrast, Abstract 3 controls the color of a light source in order to output data. Using three primary colors (red, green, and blue, for example), octal digits can be represented by a color burst containing varying amounts of these primary colors. A sequence of these signals can be used to transmit the information on a TV signal, or perhaps over a fiber-optic or other kind of optical link. The three primary color constituents are individually filtered and measured at the receiving end of the optical link. While attenuation of the signals might vary depending on wavelength in a direct optical link, the determination of simple on/off levels would seem to be quite robust. Extension to more than just eight composite colors would seem possible, although that method is not presented in the current patent. With the advent of tricolor LEDs, this scheme is intriguing for many applications.

Somewhat more conventional, it is possible to pulse-amplitude modulate a light source to encode information. Abstract 4 uses the light source to provide data and power to a cluster of low-power receiving devices placed within view of the source. An example is discussed where this system is used to present information on remote LCD displays. But remote control of all sorts is also possible.

The ability of light to couple across physical barriers while maintaining sealing is used by Motorola in Abstract 5. In this case, the unique barrier is a plastic battery compartment which contains corrosive fluids. The optical window also serves as a gas vent for the battery. One would expect that such a link might measure and communicate

Patent Number	5,115,125
Issue Date	1992 05 19
Inventor(s)	Biggs, Ross O
State/Country	AUX
US References	3,825,749 4,581,657 4,726,587 4,859,847 4,988,858 5,049,741
Title	Pin detection system with an adjustably mounted array of sensors
Abstract	A system for detecting pins of ten-pin bowling apparatus including a lens for receiving light reflected from standing pins and a linear light sensitive array for receiving light from the lens. A microcontroller is provided for controlling operation of the system and serves to provide a serial output corresponding to the level of light reflected from said pins. A housing is mounted between a pair of lanes, the housing carrying the lens and array which is adjustably mounted relative to the lens. A visual display may be provided to enable the array to be set up to receive similar light signals from each lane.



# PATENT TALK

2

Patent Number 5210604  
Issue Date 19930511  
Appl. Data 806051 1991 12 10

Inventor(s) Carpenter, Loren C.  
State/Country CA

Title Method and apparatus for audience participation by electronic imaging

Abstract An audience response system in which each audience member is provided with a reflective device which is, for example, green on one side and red on the other side. Light from a floodlight is reflected from the reflectors and imaged by a conventional video camera. A computer receives the digitized and processed image, forms a map of the audience, and distinguishes the red from the green reflections. This data is then used by the computer to project an image onto a screen visible to the audience, the image being controlled by the red or green image reflections. The audience thus jointly controls the screen image, for instance for purposes of voting or playing a game.

3

Patent Number 5182772  
Issue Date 1993 01 26  
Appl. Data 667176 1991 03 11

Assignee Karas, Christopher R., Hoch, III, Carl B.  
Inventor(s) Karas, Christopher R., Hoch, III, Carl B.  
State/Country MI

Title Method and apparatus for encoding, transmitting, receiving, and decoding information


Abstract A clear language dictionary of items of information to be transmitted is provided for composing messages and is also electronically impressed in first memory elements of sending and receiving computers. Each item of the dictionary is equated to a different octal number and the several octal numbers are impressed in second memories of both of the computers. The sending computer converts each selected item to its equated octal number, and then converts each digit of the equated octal numbers to successive groups of binary digits O-O-O to I-I-I taken from a color code which is impressed in third memories of both computers and in which the first digit of each group is equated to a first basic color, the second digit is equated to a second basic color, and the third digit is equated to a third basic color. Each group of binary digits is transmitted as a color-coded group from the third memory of the sending computer, either as a binary color group on television or as optical colors singly or in combinations of colors. Three optically shielded light-sensitive color receptors at the receiving station reconvert the color pulses to binary digit groups that are fed to the memory of the receiving computer for reverse decoding and actuation of a printer to reproduce the message sent.


battery specific gravity to the radio microprocessor. A similar arrangement could be used to communicate water temperature, salinity, or turbidity in marine applications. Or it could be used in numerous ways in sensing and communicating characteristics of process water or fluids used in manufacturing operations.

From time to time, users of the Circuit Cellar BBS have expressed an interest in measuring the speed of race cars. The patent described by Abstract 6 accomplishes just that by using a modulated laser beam emitted from one or more timing stations. The light signal is received by a photodetector within each race car and a microprocessor determines from this the exact time when the signal was received. This timing data is stored by the microprocessor and, using a polled RF system, each race car reports this time to a base station. Based on the information, the central computer can calculate various useful information such as lap time, time in corners, speed, and time in the pits.

Talking about laser light sources, there is always the need to accurately and dynamically control the operating point of a laser diode emitter. This characteristic can vary in a complex manner depending on temperature and aging of the device. Abstract 7 presents a microprocessor control system for a laser diode. One specific application presented is in an optical communication link. A number of advantages ensue. For example, optical power may be reduced until the link is completely established to avoid injury to the eyes of installation personnel. Also, the power of the laser emitter can be controlled to the minimum required for a given link, thereby reducing laser power and extending its life. It could also adjust light output to compensate for changes in the link attenuation due to fog, smoke, contamination of the windows, and so forth. By storing historical data on light output versus power input, impending diode failure can be predicted and replacement can take place without mishap.

# PATENT TALK

Patent Number	5193201	
Issue Date	19930309	
Appl. Data	512946 1990 04 23	
Inventor(s)	Tymes, LaRoy	
State/Country	CA	
Title	System for conveying a received modulated light into both power for the system and image data displayed by the system	
Abstract	A data processing system has a host computer which is coupled to a light source. The light source illuminates an area surrounding said light source and also transmits data sent to it by the host computer by modulating the light that it generates. A multiplicity of small data processing devices use photodiodes to receive both power and data from the light source. The photodiodes generate an electrical voltage which provide power to a microprocessor. A demodulator, coupled to that electrical voltage, detects peaks in the voltage which are interpreted by the microprocessor so as to regenerate the data transmitted by the light source. The microprocessor then interprets the received data and performs a task or command specified by the received data. In one embodiment, the microprocessor is coupled to an LCD display and the received data specifies what data is to be displayed on the LCD display.	

Patent Number	<b>5206097</b>	
Issue Date	1993 04 27	
Appl. Data	710421 1991 06 05	
Assignee	Motorola, Inc.	
Inventor(s)	Burns, Arthur G, Muri, David L.	
State/Country	FL	
Title	Battery package having a communication window	
	optical an	

## SOURCE

(APS) Patent Searchings e f r o m :  
 MicroPatent  
 2 5 S c i e n c e P a r k  
 1 1  
 (203) 786-5500 or (800) 648-6787

MicroPatent databases include the abstract-only APS version; FullText, which contains the entire patent without drawings; PatentImages, for the complete



*Russ Reiss holds a Ph.D. in EE/CS and has been active in electronics for over 25 years as industry consultant, designer, college professor, entrepreneur, and company president. Using microprocessors since their inception, he has incorporated them into scores of custom devices and products. He may be reached at russ.reiss@circellar.com or 70054.1663@compuserve.com.*

- 425 Very Useful
- 426 Moderately Useful
- 427 Not Useful

# PATENT TALK

6

Patent Number 5241487  
Issue Date 1993 08 31  
Appl. Data 593348 1990 10 03

Inventor(s) Bianco, James S.  
State/Country CT

Title Race car timing and track condition alert system and method

**Abstract** In a preferred embodiment, one or more timing stations disposed around a race car track. At each station, a timing signal in the form of a repeating or oscillating beam of laser light causes a photodetector mounted on a race car to turn on and off, the photodetector outputting a stream of electrical pulses. A microprocessor associated with the photodetector receives the stream of pulses, determines the real time when the signal is received, and stores that real time. When the microprocessor receives an RF polling signal unique to that race car from a base station, the microprocessor transmits the real-time data to the base station. When a second timing signal is received from the same or a second timing station, a second real time is determined, stored, and transmitted to the base station. The base station then computes the difference between the two real times. The base station processes data from all race cars in a race by sequentially polling the race cars. Different pulse rates are employed at different timing stations and recognized by the microprocessors so that lap time, total time, time through corners, and time in pit stops can be determined for each race car. In a further embodiment, there is provided an on-board track condition display responsive to signals transmitted from the base station to the race cars.

7

Patent Number 5,019,769  
Issue Date 1991 05 28

Inventor(s) Levinson, Frank H.  
State/Country CA  
Assignee Finisar Corporation

Title Semiconductor laser diode controller and laser diode biasing control method

**Abstract** A laser diode controller uses a programmed microcontroller to accurately control the process of turning on and selecting the operating point of the laser diode. The laser diode has a front facet for transmitting light and a back facet for monitoring the laser diode's optical output power. Once the back facet of the laser diode is calibrated, the controller can accurately monitor the laser diode's operating characteristics and can select the best operating point current based on the current operating characteristics of the laser diode. During calibration of the laser diode, the controller can check the linearity of the laser diode's optical output power as a function of drive current, and can thereby detect defects in the laser diode. In a full-duplex optical link, the controller of the present invention prevents the laser diodes from generating light at their full normal intensity until the integrity of the link has been established, thereby preventing light from the laser diode from accidentally damaging user's eyes. Furthermore, the controllers can use the full-duplex link to establish lower operating point drive currents that would otherwise be used, thereby significantly lengthening the lifetime of the laser diodes. A laser diode's operating characteristics change over time in such a way as to enable the controller to predict when the laser will fail. The controller records the operating characteristics of the laser diode in a nonvolatile memory, analyzes changes in those characteristics, and generates a failure warning message when those changes match predefined failure prediction criteria.

8

Patent Number 4,935,951  
Issue Date 19900619

Inventor(s) Robinson, Gary J.; Ciecierski, Walter V.  
State/Country CT  
Assignee EK-Ris Enterprises, Inc.

Title Emergency-telephone-actuated signal light or the like device and method

**Abstract** An emergency-telephone-call-actuated light signal is visible externally of the building and includes a telephone line monitor and a microcontroller for comparing the dialed numbers with a stored emergency number. If the microcontroller detects a match, it emits an actuation signal which causes operation of a light signal visible on the exterior of the building. The microcontroller includes logic for discriminating between a stored emergency number containing a series of digits and a dialed number containing the stored number sequence as a part of greater series of digits.

# CONNECT TIME

conducted by Ken Davidson

The Circuit Cellar BBS  
300/1200/2400/9600/14.4k bps  
24 hours/7 days a week  
(203) 871-1988—Four incoming lines  
Internet Email: @circellar.com

The latest news on the Internet from this month is we've added some very limited Usenet newsgroup handling. Right now we carry just sci.electronics, which generates quite a bit of traffic all by itself. We'll be experimenting with other newsgroups and more mailing lists as the months go on. If you've heard about Usenet and always wondered what it was all about, you can get a small sampling by giving us a call (when I say small, there are over 3500 active newsgroups, and the number continues to grow).

This month I decided to select just a few relatively large threads rather than a bunch of smaller ones. We often have threads that get even bigger than these. The first discussion deals with what goes into a UL-approved PC board and some of the hoops you must jump through for that approval. Next, we look at what's available for sensing tilt. Finally, we talk about CCD cameras and their use in astronomical applications.

## UL-approved PC boards

### Msg#:25638

From: JOHN MUCHOW To: ALL USERS

I have been unable to locate the UL standards pertaining to PCB layout in my local library. I am laying out my first board for an AC-powered device and would like to know the conductor spacing, widths, laminate materials, and so forth needed to exceed UL specs. Can anyone help point me in the right direction to find these specs (or a book that addresses them)? Are these specs/standards the source of the UL-94V-O (I \*think\* that's how it goes) I see on a lot of boards? Thanks for any help/tips you can give.

### Msg#:28137

From: PELLERVO KASKINEN To: JOHN MUCHOW

The 94V-O is a flammability grade and determined by UL. The other issues you seem to be looking for are not determined so much by UL as by the MIL specifications. However, there are all kinds of domestic and international rulings, partly based on human safety, partly in the fire hazard, about the spacings of "hot" lines [i.e., conductors]. But they are almost exclusively concerned about the line voltages. Anything beyond the transformers or other sources of energy are not of concern to the same codes.

At most, the agency practices are to make a short circuit during testing on anything that is closer than about

$\frac{3}{16}$ " to each other and if a fuse clears the short circuit without causing a fire, that's acceptable to them. There have been signs of changing attitudes and push towards more detailed "robust" behavior in some cases, but that does not provide clean rules for most board designers, as the MIL specifications do.

One source of information about these is a book that we bought several years ago from Bishop, the makers of precision slit tapes and other PCB design materials. (What realization: For several years now I have used a computer and laser photoplotting services—not those tapes! In fact, we have only used them for some front-panel artwork.. .) The names of the book and the author now escape me; should have checked at work before I came home to enter this message. Sorry.

### Msg#:28207

From: JOHN MUCHOW To: PELLERVO KASKINEN

Thanks for the 94V-O info. I guess each company determines, on their own, whether their board meets UL specs for flammability and places the 94V-O code on the board? Or do they have to prototype the board, send it to UL for testing, and then get the OK to produce additional boards with the code? It seems that UL wouldn't want to have to test every PCB coming through its labs, so they would trust the manufacturer to do its own proper research. UL would just continue to do "systems" testing, not component testing, of each product. Unless, of course, a manufacturer was submitting just a component for approval.

It's interesting that Bishop had that information in their book! If you do come by the name/author of the book, I would appreciate it if you could let me know. My next step seems to be to contact UL, and probably ETL too, to see if they have anything they can send. Thanks for your help!

### Msg#:29236

From: PELLERVO KASKINEN To: JOHN MUCHOW

I did take a look at the Bishop book and noticed that it was not exactly what I had claimed. Must be that I got the spacing requirements from some other source after all. But like I said, all of that comes from the MIL standards originally. I also remember there was something about



# CONNECT TIME

altitude corrections. If the board was intended to be flying at high altitudes, the spacing had to be increased.

Now, back to the flammability. It is a material issue, not layout or component question. If you have picked the appropriate board material, you've got the issue settled without having to send for UL approval. Actually, it is like the UL-listed issue versus UL-recognized component. When a board material supplier has the material in the recognized category for the flammability class, then it does not have to be retested. Of course, the general safety is determined by various other details.

I think I may have pointed to the difference between European and the American safety rules. While both cover the fire hazard and shock hazard, it appears that the American rules emphasize the fire prevention, while the European rules elaborate much more on the shock side. Maybe due to the factor that the main sponsor for the National Electric Code is the National Fire Protection Association.

Back to the Bishop book: it was written by Darryl Lindsay. Our copy is from 1984, in the second edition, third printing. ISBN 0-9601748-1-8. And at least then Bishop Graphics was at 5388 Sterling Center Dr., Westlake Village, CA 91359.

Sorry for the lack of more firm numbers. I still cannot remember where I actually have seen the tables of the trace spacings. May even have been some magazine article "anno dazumal" or once upon the time..

## Msg#:30490

From: JOHN MUCHOW To: PELLERVO KASKINEN

Thanks for getting back to me on the Bishop book; I'll investigate and let you know how it went.

Our boards are all 62-mil FR-4, so we should have no problem there. We've taken the voltage spacing recommendations of Printed Circuits Design and Printed Circuits Handbook and doubled them so \*that\* shouldn't be a problem either. Looks like the next step is calling UL and ordering the appropriate standards (we finally received their "Standards for Safety" catalog and \*ouch\* they're expensive!), studying and applying them where necessary, and then take the plunge and start a UL investigation. Ooh boy, this stuff is scary the first time around! Thanks again!

## Msg#:30933

From: PELLERVO KASKINEN To: JOHN MUCHOW

Regarding the UL prices-just wait until you see the actual inspection fees! And they want to make it a permanently recurring occasion of visiting your factory (to see that you stay in conformance..). Of course, you pay their visit.

You might want to get some "second opinion" from some people that do miscellaneous compliance engineering. I think their services may be slightly more affordable and also they would offer viewpoints regarding the international codes as well. I believe there are about four or five such services.

## Msg#:31110

From: JOHN MUCHOW To: PELLERVO KASKINEN

I've spoken to a few people recently and they all say the same things you've brought up; price!, price!!, price!!! A company I've done some consulting for, a producer of tungsten lights, highly recommends ETL. I'll be contacting them soon to find out what they cover and their price. Sigh.. it would be so nice if you could just use all UL-listed components and that would be the end of it <g>. Thanks for your feedback and advice, it's always appreciated.

---

## Tilt sensors

### Msg#:29590

From: TODD WENZEL To: ALL USERS

Does anyone know of any manufacturers of solid-state devices which can show the degree of "tilt" from horizontal or vertical?

I remember seeing a "smart" level from Stanley (or someone) which gave some sort of digital indication of degrees from horizontal or vertical. I could really use a transducer like this for a project I'm working on. If you have any ideas, I would appreciate any direction you could point me in.

### Msg#:30012

From: ERIK KAUPPI To: TODD WENZEL

Yes, Stanley makes one. I have seen similar items with other brands, don't know if they are the same inside or not. I don't know what Stanley is using inside their unit.

One technology that is often used for this sort of thing is an electrolytic transducer. The brand name I know is Fredericks; they advertise in Sensors magazine. The sensor is a glass vial with electrodes and a liquid inside it. It needs AC excitation and outputs a differential AC signal depending on how much of each electrode is in the liquid (or something like that).

### Msg#:30180

From: PAUL PETERSEN To: TODD WENZEL

One other area that you should peruse is a device known as an Inclinometer, which is used in marine (ship-

# CONNECT TIME

board) applications to measure pitch and roll. The inclinometer is essentially a potentiometer inside a sealed steel box. You read the wiper on the pot; there are two of them at right angles for pitch and roll. Sorry i don't recall the name of the manufacturer, it's been a lot of years.

**Msg#:30332**

**From: JIM NELSON To: TODD WENZEL**

Try Spectron Glass and Electronics, (5 16) 582-5600.

Spectron manufactures a bunch of different tilt sensors-basically liquid potentiometers. The high-sensitivity versions look exactly like the vials in carpenter's levels, but with three wires secured via epoxy. Their wide-angle (up to  $\pm 60^\circ$ ) sensors look like tiny, squashed vacuum tubes and sense tilt in two dimensions. The smallest sample we could get is just over 0.6" in height from the PCB surface mounting plane and about 0.5" in diameter.

The Fredericks Company (215/947-2500) makes one that is good for 360" with the same glass and electrolytic sensor technology used by Spectron. Linear Technology App Note 43 has a section on using these sensors in AC bridges.

I haven't noticed a solid-state version of this technology. Amp does make a very, very small surface-mount accelerometer which allows you to choose to sense either rotation around a single axis or translation in one axis by varying the connections to the pair of op-amps included in the package. You would need perfect integrators to track tilt or position this way. Lacking these, coupled with the (Pennwalt) piezoelectric plastic's lack of very low frequency response, makes this discussion somewhat moot.

Analog Devices makes a monolithic accelerometer, including signal conditioning, but again with only one dimension of acceleration sensed. Connect a couple of these differentially, in the same manner as is used in the Amp unit, and you could sense rotation.

Both manufacturers claim that their sensors will be in the \$5 ballpark at million-unit volumes-next year.

I've only got to recalibrate my "Smartlevel" once in a while. So my SWAG is that it uses the electrolytic pot technology. Any accelerometer-based technology would require recalibration after every power-down.

**Msg#:31771**

**From: PELLERVO KASKINEN To: TODD WENZEL**

You already got some replies. I think I can only emphasize some detail and add one source. The source for most any position and angle transducer that I have ever needed is Schaevitz. If i remember correctly, it is now part of some concern like Lucas. Anyway, they have transducers based on the LVDT principle, both linear distance and angle transducers. The transducers are "industrial strength" and

probably in the \$100+ arena. You also need the AC signal conditioner to work with them.

What I plan to elaborate is the solution of using a small instrument-type potentiometer. Mostly the potentiometers in that type come with ball bearings and a very low friction. They also have what is called a servo mount, a ring all around the body instead of the normal threaded bushing. This ring allows you to gently rotate the potentiometer before locking it in place to achieve position calibration. To mount the device, you would have at first a large hole matching the corner of the pot body. You push the pot in and then use L-shaped grippers that rest against the panel with the shorter leg and the longer leg goes into a groove in the pot body. There are small holes on the longer leg of the L and you put a tiny screw through it to the panel. Normally you would use three grippers.

Now, all this is just a precursor. You would need to build a bridge circuit, adding a trimmer potentiometer for zero offset. The measurement then takes place between the two wipers. There is, of course, a need for something that actuates the detector. It would likely be a simple pendulum attached to the shaft of the instrument potentiometer.

And then you have to calibrate the device? Select a sturdy horizontal table top. Put your level on there and adjust the trimmer potentiometer to get a first zero reading. Turn the level exactly  $180^\circ$ , end to end. As your table is unlikely to be perfectly horizontal, you now get a small reading. Divide that by two. Adjust the trimmer potentiometer until your meter reading matches this half of original. Lock things in place. Double check by again rotating the level end to end. Should be reading the same, though opposite polarity. Done!

---

## CCD use with telescopes

**Msg#:18155**

**From: GREG RIDDICK To: ALL USERS**

I am currently trying to construct a low-cost, low-resolution CCD camera. I'm looking for CCD devices, hopefully surplus, that could provide 100x100-pixel resolution or less for under \$100. There seem to be a few single-line CCDs out there from hand scanners that are very reasonably priced, but that would require some kind of linear scanning mechanism (possible, but a little cumbersome). Has anybody out there found a good source for CCDs or had any experience working with them? Any help would be greatly appreciated.. .

# CONNECT TIME

**Msg#:18212**

From: JAMES MEYER To: GREG RIDDICK

Contact Sanyo. They are the company that made the CCD for the Fisher-Price "Kiddie Camcorder."

**Msg#:18432**

From: GREG RIDDICK To: JAMES MEYER

Thanks for the tip, James. Will try to contact Sanyo. I am also checking out Texas Instruments at the moment, but their CCDs look very pricy.

**Msg#:21544**

From: DAN COTE To: GREG RIDDICK

For information about CCDs, you might try two books written by Richard Berry on CCD cameras and image processing. Both are published by Willmann Bell Inc., P.O. Box 35025, Richmond, Virginia 23235, (804) 320-7016. Berry's first book, "Astronomical Image Processing," was published 2-3 years ago. The book is made for newcomers into image processing. The book also comes with a program named "AIPro" and few samples images. The software is made for proprietary CCD camera file formats and the source code in Quick Basic is included. This is a good book for those who own one of the supported CCD cameras. It is just as good for those who want to learn the basics.

His second book came out last year and it's named "Choosing and using a CCD Camera." This is a practical guide to getting the most of commercial CCD cameras sold for imaging with a telescope. This book also comes with a special program named "QuickPIX" which is a subset of AIPro but much faster and easier to use (no source code).

I met Richard Berry last August in Springfield, Vermont at the annual Stellaphane Astronomy Convention. He showed me his latest "baby" that is a homemade CCD camera based on Texas Instruments' TC245 CCD chip. He confirmed that he was working on a book on homemade CCD cameras. His book will provide all the information needed to build two models. One will be a TC211-based camera while the other will use the TC241 or the TC245. Berry said that both models will be sold as a kit by University Optics. The book will be published by Willmann-Bell and it should be available by now.

The prototype I saw was working very well. It was even as good as the SBIG ST6 CCD camera that is sold for \$3000 US (\$4800 in Canada). He said that all parts should cost around \$400-500 for the big model and less than \$200 for the small one. Both models are temperature regulated (cooled) by a Peltier cell with the hot side cooled by water circulating in an aquarium-type pipe. This could be changed by a heatsink, though.

I use SBIG ST4 and ST6 CCD cameras when weather permits. These aren't mine, and I'll wait for Berry's book before I spend a cent on a CCD camera. By the way, I wrote an image processing program named "TIAstro." It supports Lynxx, Electrim and ST4 CCD formats. It is available as a shareware. I wrote that few years ago for my own use and I did an English translation last summer (my main language is French). I'm working on a Windows version now and it will support GIF, TIFF, TARGA, PCX, and new formats from CCD manufacturers. If there's an interest for you or others, I'll upload it here. I'm an avid reader of *CCI* and I plan to log here once a week or so.

**Msg#:21752**

From: BARRY KLEIN To: DAN COTE

I'll be looking forward to the coming book.

I really had to make the TI sales rep go through hoops to get me a TC-211 sample. Glad to see someone will have kits available. I plan to get a basic circuit functional just to become familiar with the technology and then experiment with larger arrays and alternative drive/AD circuits. I am also looking forward to the new **CCD** magazine that will be coming out early next year. I'm sure we'll see some homebuilt designs there, too.

**Msg#:22757**

From: DAN COTE To: BARRY KLEIN

I know a guy that made a TC211-based CCD camera from an article in **Telescope Maker** magazine. He also had it at Stellafane last August. I could find a local supplier for the CCD chip and I would be glad to see what I can do from the **TM** article. I have subscribed to the new **CCD** magazine from Sky & **Telescope**. We did a trichrome image of M57 using a ST4 CCD and we have sent it to Dennis Diccio for a future publication in *S&T* or **CCD** magazine. I will try to implement the clocking pulses on a microcontroller instead of using a PC link. This will be a fun project for this winter!

**Msg#:22844**

From: BARRY KLEIN To: DAN COTE

Let me know how you make out with the microcomputer-controlled CCD project-I had the same idea. I would also like to know more about your source for parts. I may like to try the larger parts sometime in the future (can they get them and do you have any idea for pricing? ).

**Msg#:23602**

From: DAN COTE To: BARRY KLEIN

I bought a book from the publisher of *S&T* named "CCD Astronomy" by Christian Buil. This is a great book split in two major parts. The first deals with electronics while the other is on image processing on a PC.

# CONNECTIME

Buil describes how to build a thermoelectric-cooled CCD camera, but the design uses a Thompson CCD chip which is hard to find around here (Buil is from Belgium). David Groski (a friend) used the design from *TM* and he added features from the Buil's design. Richard Berry used some of David's designs for his book. Actually, Berry didn't design any of the two models that will be in the book. He picked ideas here and there and he hired professional engineers to make the final design, and I think he made a good decision because he wanted the best possible design. He wrote himself the acquisition software and he told me that he will add a track-and-accumulate feature like the one found on the SBIG's ST6. Berry's software is better because the CCD can guide the scope while it does imaging. How he does it? I can't tell.. .

I bought the Basic Stamp development kit from Parallax and I'll check if it could be used as the main controller. Maybe I'll need more than a Stamp to do it. At the moment, I'm getting familiar with the language, which is easy to learn. I'm testing it with stepper motors for something that will become an automated CCD camera tracker. I have modified a voice-operated telescope design that was published by CCI some time ago. The UCN5804B chip makes it easier to control steppers.

Last winter, I built digital setting circles using a optical encoders and a binary counter. The system is linked to the PC via the parallel port. I never installed it on my scope because I want it to work as a stand-alone unit with a serial PC link. Again, the Stamp will take care of the logics. I still don't know if the Stamp is fast enough to control the CCD. David told me that anything that could pulse a line at a resolution of 10  $\mu$ s should do the job, and 10  $\mu$ s is the theoretical limit of the Stamp. I'll let you know when I find it out.

I have the same problem as you with parts. I found few TC211 at Future Electronics in Montreal. These are listed at \$75 CND (\$55 US). You may want to call Texas Instruments at (800) 336-5236 to find your nearest distributor. CCD chips from TI shouldn't be too hard to find. David also uses two TMS3473B as a CCD chip driver. He ordered these directly from TI. The line drivers to the PC parallel port are two 74HC240s, which can be found at any parts supplier. The cooling system could be made of a Peltier cell and a 0P741 set up as a comparator and a cheap thermistor. I can't remember which A/D converter he used. I know this is a 12-bit part, which is much better than the SBIG ST4 at 8 bits. David's camera has six chips in all.

David recommended to me to read *Observatory Techniques* magazine, Summer '93, where his design is discussed (I think he said they published his design). The Fall '93 issue should present the software to control the CCD camera. There's also a company named CCD Tech-

nology who runs a BBS. David got parts and CCD kits from them. I don't have their address but the BBS's number is (714) 633-6985.

Few days ago, I got a message from David about a new CCD camera he just built. He used a TC245 CCD (786x488). He told me that the first one is the hardest to build. I guess this is true since he did this new camera in just a few weeks. The big one is as good as the \$3000 priced ST6 and it cost about \$350 in parts.

Finally, I'll do my CCD camera from an existing design and then modify it to be controlled by a microprocessor like the other parts of the automated tracker. I'm also thinking about building a serial radio link between the tracker and the PC. Any suggestion for a good emitter/receiver? I'm looking for something about the size of a pack of cigarettes. The radio link would permit me to control the tracker/CCD from a car or my home computer. At  $-35^{\circ}\text{C}$ , these are the places to be...

## Msg#:28273

From: BARRY KLEIN To: DAN COTE

At this point, I have put together the ATM circuit on the board I bought and plan to run 8 wires out (6") to the CCD module. I am wondering if I can get away with it—without including the output buffer, trimpot, and so forth in the CCD module. I bought a surplus IR laser diode assembly a while back that I am modifying to be the CCD module. It has a built-in lens (that I'll take out) and Peltier cooling module. I took out the laser diode and put the TC211 in its place. The cooling module is nearby on the metal casting.

I am curious what approaches you have seen used for the cooling hardware. I am worried about frosting of the CCD window. I read somewhere that the cooling module may attract the frost (humidity) if it is in the same air environment. A guy at work suggested I run CO<sub>2</sub> gas into the chamber to prevent frosting. I may do this or get another Peltier module to collect frost away from the environment. The assembly I have has a large heatsink on the back, but I may have to use liquid cooling on it. What have you seen done in these areas? If I have to have a vacuum chamber, I'm afraid that's where I give up as the expense and effort for me will not justify it. I have the Buil book and have read of his efforts (by the way).

## Msg#:30337

From: DAN COTE To: BARRY KLEIN

You're lucky to be that far! All I did to date is to read the ATM article twice. I'm quite busy these days [read: work and work!].

I've seen few different cooling systems on commercial and professional CCDs. The worst (hard to build) was

# CONNECT TIME

certainly a cooling system using liquid nitrogen. The easiest approach would be to do a hole on the circuit board under the CCD chip and stick a copper rod to the back of the chip. Here's how David Groski explained his method to me:

"To interface the CCD to the coolers, I used a piece of  $\frac{5}{16}$  copper rod that is about  $\frac{5}{8}$  long. The rod was filed square so it fit between the pins of the CCD. The top of the rod was polished flat with 1000-grit sandpaper, the bottom is soldered to a flat copper plate that is the same size as the first TEC. The plate was also sanded flat with 1000-grit paper.

"Both surfaces of the copper cold finger have a light coating of heatsink grease on them. The rod passes through a hole in the PCB. The CCD is enclosed in a 1.25 heavy-walled brass tube with a microscope slide glued to the top for a window. The bottom of the tube has a brass plate soldered to it that extends out around the tube. I duct taped the plate to the PCB. The inside of the tube is lined with black foam and so is the outside of copper cold finger and any exposed surface of the TEC. The back of the TEC is greased and is against the heatsink.

"As for thermistor, I used a Radio Shack one, about \$1.50 each, and they come with a table of resistance versus temperature over 200°C so I could easily fit a curve to it."

I don't think frosting will be a major problem if the CCD chamber is made of a metallic material. When the chamber gets to the dew point temperature, the humidity will stick to the inside of the chamber and then freeze when temperature drops. As long as the chamber is cold, the humidity will stay on the side.

Richard Berry uses a water cooling system. He wraps a small copper pipe around the heatsink. Each end of the pipe is connected to a little aquarium pump with a transparent rubber pipe through to make water circulate in a loop. Someone told me that using windshield washer fluid would prevent getting some "alive" particles in it. Berry's system also has an 8" spiral coil made from the rubber pipe and he puts the coil in a water bucket. He could get his CCD to a temperature lower than -30°C while it was +25°C outside.

The vacuum chamber would make sure you don't get any frost. David didn't have any problem with his system to date. He used large heatsinks and both of his cameras are air cooled. He can reach around -25°C, which is good.

In my opinion, the Buil book is a good one. I have the original French edition and the American edition. I didn't look at the new edition by Willmann-Bell yet. By the way, David just got a prepress copy of Berry's book. He says Berry did a wonderful job of explaining it all. Since Berry did many experiments to build his CCD, the book is full of tips for those who would like to improve the suggested design. Willmann-Bell now says that the book will be available in mid-February. I'm waiting anxiously for it.. .

## Msg#:28418

From: MICHAEL MONTEITH To: BARRY KLEIN

I have a TC211 and I'm interested in seeing how you did your electronics. It's been a long time and I haven't dug out all my stuff on it, but I always like to hear how someone else did it.

As far as the cooling devices, Marlow Industries sells the device relatively cheaply for the size you need instead of stripping other assemblies. I would love to have had the laser diode assembly. Anyway, my suggestion is to use nitrogen instead of CO<sub>2</sub>. Nitrogen is used in many high-voltage and lenses where moisture is a concern. If you also make the pressure inside relatively low, you will not have to worry about having the plate covering the whole unit frosting from thermal transfer from the temperature inside.

## Msg#:32347

From: DAVID MEED To: BARRY KLEIN

Has anyone has used a vortex cooler for cooling? Don Lancaster has mentioned it in his columns in *RE*. Basically it is a vortex (like a snail shell) into which you put compressed air and it puts out (very?) cold air on one side and hot air on the other side. Unfortunately it has been a while and I can't remember how cold it gets, and of course you need a source of compressed air.. .

---

*We invite you call the Circuit Cellar BBS and exchange messages and files with other Circuit Cellar readers. It is available 24 hours a day and may be reached at (203) 871-1988. Set your modem for 8 data bits, 1 stop bit, no parity, and 300, 1200, 2400, 9600, or 14.4k bps. For information on obtaining article software through the Internet, send Email to [info@circellar.com](mailto:info@circellar.com).*

## ARTICLE SOFTWARE

Software for the articles in this and past issues of *The Computer Applications Journal* may be downloaded from the Circuit Cellar BBS free of charge. For those unable to download files, the software is also available on one 360K IBM PC-format disk for only \$12.

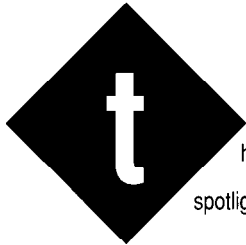
To order Software on Disk, send check or money order to: The Computer Applications Journal, Software On Disk, P.O. Box 772, Vernon, CT 06066, or use your VISA or Mastercard and call (203) 875-2199. Be sure to specify the issue number of each disk you order. Please add \$3 for shipping outside the U.S.

## IRS

428 Very Useful      429 Moderately Useful      430 Not Useful

# STEVE'S OWN INK

And the Survey Says...



Here's good news and bad news about being an old timer in the computer industry. The good parts primarily spotlight the memories of being there when today's computer moguls were still arguing about being in cramped garages or having to book orders themselves. I generally try not to publicize that I thought a junkie-looking prototype of the Apple I would go nowhere or that a \$49 Pascal might not be taken seriously. Of course, those missed stock opportunities are mitigated by memories of being with the key players at critical junctures and witnessing contracts being written on napkins during West Coast Computer Faire and National Computer Convention press receptions (before COMDEX became the big one). I can still remember those orange-size chocolate-dipped strawberries at one Microsoft party....

Being with BYTE in the early days gave me an opportunity to view both computer publishing and computer manufacturing from infancy to maturity. Probably the single most profound deduction from that experience was, "know thy customer." When Carl Helmers started BYTE and I was writing all those construction projects, we each had a specific image of the reader in mind (his was a physics major who read *Scientific American*, while mine was an EE with exactly my interests). Fortunately our ideas were synergistic.

I had the same editorial focus when Circuit Cellar's *Computer Applications Journal* was started. Its success suggests that either we've targeted the audience correctly or there's something particularly special about the paper we use that cats like. Of course, the only way to truly know these answers is to ask, so we recently sent editorial questionnaires to 2000 CAJ subscribers.

The results were indeed enlightening. The very high response, regardless of whether we provided return postage or not, demonstrated that CAJ readers, like the original core audience of BYTE, want to play an active role in our editorial focus. They didn't overlook a single question or complain about answering four pages of detailed questions. Many respondents took the opportunity to provide volumes of supplemental information and comments.

As you probably guessed, ours is a highly technical audience directly involved in microcomputer-based applications and its associated software. Though evenly divided between small and large company affiliation, CAJ readers own multiple computers (75% say they use Windows software on the job), are well educated (97% have been to college, 46% have more than a college degree), and make lots of money (65% make more than \$50k). Nearly 60% of our readers have been with us for five years or more. As far as they are concerned, they are core audience "lifers."

Finally, the financial types wouldn't let me leave without stating certain demographics important to them. CAJ readers are action oriented. 78% responded that they are regularly involved in product and/or vendor selections made by their company. 98% say they read all the ads, while, at the same time, 93% have contacted advertisers and/or purchased a product as a result.

Well, my vision of the CAJ reader still stands. Like the computer industry we serve, we are a bit older, a bit wiser, and a lot more opinionated. CAJ will continue to focus on useful technical stuff, albeit sometimes with a sense of humor, but don't expect any toaster or appliance reviews anytime soon.